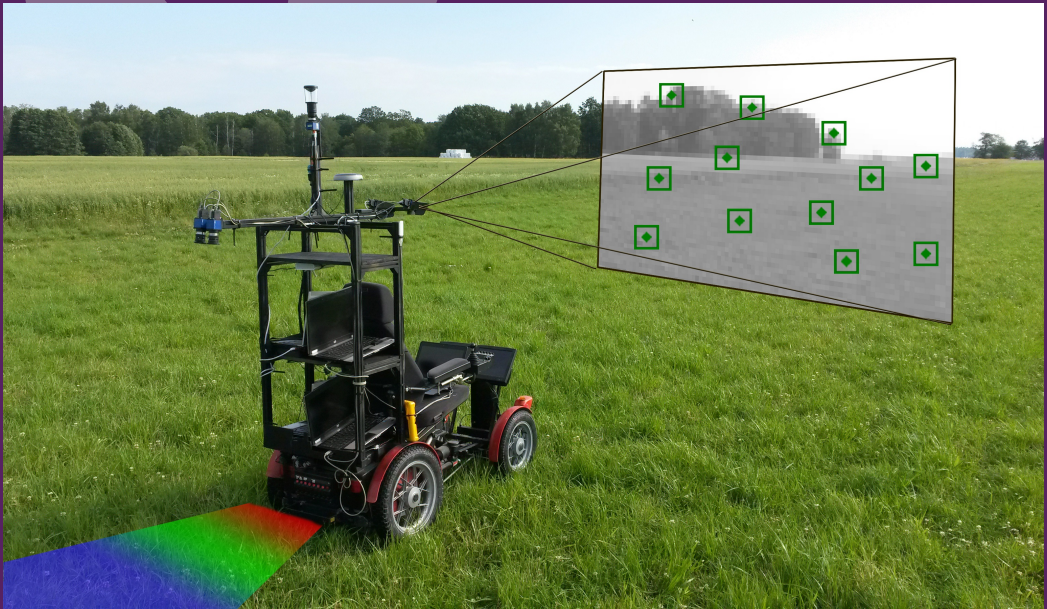




UNIVERSITY
OF SKÖVDE



DOCTORAL DISSERTATION

VISION-BASED PERCEPTION FOR LOCALIZATION OF AUTONOMOUS AGRICULTURAL ROBOTS

STEFAN ERICSON

VISIT...

LANZAROTE
Caliente.COM

STEFAN ERICSON

VISION-BASED PERCEPTION FOR LOCALIZATION OF AUTONOMOUS
AGRICULTURAL ROBOTS

2017

DOCTORAL DISSERTATION

VISION-BASED PERCEPTION FOR LOCALIZATION OF
AUTONOMOUS AGRICULTURAL ROBOTS

STEFAN ERICSON
Informatics



Stefan Ericson, 2017

DOCTORAL DISSERTATION

Title: Vision-based perception for localization of autonomous agricultural robots

UNIVERSITY OF SKÖVDE, 2017, SWEDEN

www.his.se

Printer: Runit AB, Skövde

ISBN 978-91-982690-7-9

Dissertation Series No. 16

To my family

ABSTRACT

Industrial robots have been used for decades in industry, which has resulted in fully automated production lines. The automation level of field operations in agriculture, however, is much lower, even though there are tasks particularly suited to automation. One example is weed control in organic vegetable production, which is still to large extent done manually. While autonomous robots used in research can carry out agricultural tasks like weed control, there are still technical challenges relating to reliability and robustness. Robustness involves the ability to operate in a highly dynamic agricultural environment, where dirt, moisture and weather have to be taken into account. The process of mechanical weeding, for example, can be dirty, and so any cameras used for plant identification have to be mounted far away from the tool. This then requires the position of the tool to be measured relative to the identified weed.

This thesis demonstrates how cameras can be used for localization of an agricultural field robot. Simulations, laboratory experiments, and real field experiments were done to evaluate different ways of localizing a mobile robot on an agricultural field. The simulations generated images similar to those expected in the field, and these images were used to evaluate specific parameters or design choices relating to the algorithms used. The laboratory experiments targeted evaluation of different ground textures on a small scale, while the real field experiments included data from driving a mobile robot several kilometers in an agricultural field.

One method of measuring the relative position of a camera is to track changes between consecutive frames, a procedure known as visual odometry. Visual odometry was evaluated for localization on different camera setups including downward-facing and forward-facing cameras. The localization can also be done while building a map of the environment in what is known as simultaneous localization and mapping (SLAM). SLAM was also evaluated in terms of global estimation of the robot's position on the field. The cameras can also be used to identify structures such as rows in the field, and the robot can be programmed to follow these rows.

The results show that a 2-degree-of-freedom visual odometry system can be used with a downward-facing camera to estimate relative position over short distances, as when estimating the position of a tool relative to a crop identification camera. A real-time implementation of this visual odometry system can also be used in combination with a row detection system to autonomously control a mobile robot so that it follows a row and know its position. The row detection system measures the robot's position relative to the rows by using images from a forward-facing camera. A demonstration run showed that when the robot drove autonomously along a 10 meter track, the position error at the end position was 2% of the travelled distance. It was also shown that the row detection system can be improved by using an omnidirectional camera and an algorithm that detects rows that are pictured as both individual plants and solid lines. This system can also detect the vanishing point of row-structured fields.

Visual odometry and SLAM can be used for localization with 6 degrees of freedom, i.e. position (x, y, z) and orientation (θ, ψ, φ) , in an agricultural field. Forward-

facing stereo cameras were more accurate than downward-facing cameras. The results from the simulations show that high resolution images from stereo cameras with a wide baseline provide increased accuracy. The main source of error is drift, which occurs because of the accumulation of errors in each frame analyzed. Errors can also occur when either too few or too many frames are used for localization. The use of the SLAM algorithm overcomes the problem with frame rate selection since both local and global optimization of the position are performed. This algorithm also has the ability to close loops, but due to the traditional back-and-forth driving pattern no loops were closed. That led to drift, especially in heading angle, so that this approach showed similar performance to visual odometry using an adapted frame rate. The best localization accuracy was obtained on a 2.4 kilometer run using the algorithm ORB-SLAM with forward-facing stereo cameras. The median error of running the algorithm five times on the same data set showed a translation error of 2.63% and 0.0321 deg m⁻¹ orientation error. This result shows that it is possible to use cameras for localization on an agricultural field.

This approach could be used to improve the robustness and reliability of existing systems, which could potentially enable commercial autonomous field robots to perform agricultural tasks in an economical and ecologically sustainable way.

SAMMANFATTNING

Inom industrin har robotar används i årtionden för automation, vilket har resulterat i helt automatiska produktionssystem. Automationsnivån inom jordbruket och då främst av uppgifter som utförs på åkrarna är mycket lägre, trots att det finns uppgifter som är särskilt lämpliga att automatisera. Ett exempel är ogrärensning i ekologiska växtodlingar, en uppgift som idag till stor del sköts manuellt. Trots att det finns autonoma forskningsrobotar som kan utföra ogrärensning finns det fortfarande tekniska utmaningar i att göra dessa pålitliga och robusta. Med robust avses här förmågan att kunna arbeta i den dynamiska jordbruksmiljön där roboten måste klara av att arbeta i smuts, fukt och olika väder. Mekanisk ogrärensning till exempel är ofta väldigt dammigt, så eventuella kameror som används för att identifiera ogräset måste monteras långt ifrån rensningsverktyget. Detta i sin tur kräver att verktygets position kan bestämmas relativt de identifierade ogräsplantorna.

Den här avhandlingen visar hur kameror kan användas för positionsmätning på ett jordbruksfält. Simuleringar, laborationsexperiment och fältförsök har används för att utvärdera olika sätt att mäta positionen av en mobil robot på en åker. Simulatorer har används för att skapa bilder liknande de som förväntats i fält och dessa har används för att utvärdera specifika algoritmparametrar och designval. Laborationsexperimenten har använts för att utvärdera olika markstrukturer vid positionsbestämning på korta avstånd, och fältförsöken som utförts på riktiga åkrar innehåller data från flera kilometer långa körningar och har använts för utvärdering av lokaliseringsalgoritmer på åkern.

Ett sätt att mäta relativ kameraposition är att följa ändringar mellan två bilder, så kallad visuell odometri (eng. visual odometry). Visuell odometri utvärderas i avhandlingen för lokalisering med olika kamerakonfigurationer såsom nedåtriktade eller framåtriktade kameror. Lokalisering kan även utföras genom att samtidigt skapa en karta över omgivningen, så kallad SLAM. På så sätt kan robotens globala position på fältet mätas. Kameran kan även användas för att detektera olika mönster i fältet, till exempel rader. Roboten kan sedan programmeras för att följa dessa rader.

Resultat från avhandlingen visar att visuell odometri kan användas på nedåtriktade kameror för att mäta den relativa positionen på korta avstånd i två frihetsgrader. Detta kan användas för att estimerar positionen av ett verktyg relativt till exempel en ogräskamera. Genom att implementera algoritmen för realtid i kombination med ett system för raddektering kan roboten autonomt följa en rad samtidigt som det är möjligt att veta robotens position. Raddetekteringssystemet mäter robotens position relativt raderna med hjälp av bilder från en framåtriktad kamera. En demonstration visar att roboten kan följa en rad utefter en 10 meters bana och positionsfelet mätt vid slutpositionen är 2% av körd sträcka. Försök visar också att raddetekteringssystemet kan förbättras genom att använda en runtseende kamera (med synfält på 360°) och en algoritm som klarar av att detektera både individuella plantor och hela rader. Systemet kan även mäta positionen på flyktpunkten av ett fält med rader.

Visual odometry och SLAM kan användas för lokalisering med sex frihetsgrader, dvs. position (x, y, z) och riktning (θ, ψ, φ) , på en åker. Framåtriktade kameror

ger bättre noggrannhet i positionen jämfört med nedåtriktade kameror. Resultaten från simuleringarna visar att en högre upplösning och ett längre avstånd mellan stereokameror ger en ökad noggrannhet. Den största felkällan är drift som beror på att fel från varje analyserad bild ackumuleras. Både för hög och för låg bildfrekvens orsakar ökat fel. Användning av en SLAM algoritm löser problemet med att hitta optimal bildfrekvens då både en lokal och global optimering av positionen utförs. SLAM har även förmågan att detektera och korrigera fel då roboten återkommer till en tidigare besökt position. Detta var dock inget som inträffade under något av fältförsöken där roboten kördes fram och tillbaks på en åker. Detta ledde till drift av främst orienteringen, vilket resulterade i en felaktighet i positionen liknande den som visuell odometri gett. Bäst resultat uppnåddes på ett 2.4 km försök genom att använda algoritmen ORB-SLAM på de framåtriktade kamerorna. Medianfelet för fem analyser med algoritmen visade på ett positionsfel på 2.63% av körd sträcka, samt 0.0321 grader per meter i rotationsfel. Det här resultatet visar att det är möjligt att använda kameror för lokalisering på en åker.

Den här metoden skulle kunna användas för att förbättra robustheten och pålitligheten hos existerande system, vilket också skulle kunna möjliggöra kommersiella autonoma robotar som kan utföra jordbruksuppgifter på ett ekonomiskt och ekologiskt hållbart sätt.

ACKNOWLEDGEMENTS

This dissertation marks the end of a long journey that began as a dream of getting a PhD. The topic that caught my attention was how to use camera images to control a mobile robot. My background, however, is in power electronics, specializing in designing motor controls and power supplies, so there has been a lot of new knowledge to gain. I would like to thank Klas Hedenberg, who shared this journey with me, especially the joint effort of making the experimental platform run properly. I would also like to thank the municipality of Mariestad for giving me permission to use their agricultural fields.

This work would not have been possible without the support of my supervisor, Björn Åstrand. Thanks for staying up late at night to support me when writing papers close to deadlines. I also would like to thank Anna Syberfeldt for making me work harder than ever to complete this thesis. You also provided the support that I needed to make this possible.

I am also grateful to others who played an important role in this work. Thanks to Tom Ekblom for sharing your programming skills. Your way of structuring problems and your instructions on what to do and not do when programming saved me hours of debugging. Thanks to Henrik Smedberg for your programming support and for running the algorithms over and over again, providing important results for this work. Another person who deserves my appreciation is my brother Mattias Ericson who contributed the artistic design on the front page. I also received invaluable feedback from David Vernon. Thank you for interesting discussions and for sharing your wisdom.

Finally, I would like to express special thanks to my wife, Irina Ericson, for being there for me. You listened to me for hours when I was talking about things you probably didn't understand. Yet you were still able to give me valuable advice and put things in a completely different perspective. You also supported me by giving me time to work. Without you this work would not have been possible.

LIST OF INCLUDED ARTICLES

In the course of my studies, I produced the following publications, which have either been published or are currently under review.

PAPER I

Stefan Ericson and Björn Åstrand (2007). “Algorithms for Visual Odometry in Outdoor Field Environment”. In: *Proceedings of 13th IASTED International Conference Robotics and Applications*. Würzburg, Germany, pp. 287–292

PAPER II

Stefan Ericson and Björn Åstrand (2008). “Stereo Visual Odometry for Mobile Robots on Uneven Terrain”. In: *IAENG Special Edition of the World Congress on Engineering and Computer Science, Advances in Electrical and Electronics Engineering*, pp. 150–157

PAPER III

Stefan Ericson and Björn Åstrand (2009). “A vision-guided mobile robot for precision agriculture”. In: *Proceedings of 7th European Conference on Precision Agriculture*. Wageningen, Netherland, pp. 623–630

PAPER IV

Stefan Ericson and Björn Åstrand (2010). “Row-detection on an agricultural field using omnidirectional camera”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010. IROS 2010*, pp. 4982–4987

PAPER V

Stefan Ericson and Björn Åstrand (2016). “Ego-motion estimation by an agricultural field robot using visual odometry”. Submitted to Biosystems Engineering

PAPER VI

Stefan Ericson and Björn Åstrand (2017). “Localization of agricultural field robots using natural landmark and vision SLAM”. Submitted to Sensors

CONTENTS

1	Introduction	1
1.1	Motivation	1
1.2	Aim and objectives.....	4
1.3	Research questions	4
1.4	Relationship between Articles	5
1.5	Main contributions of the included articles	6
1.6	Outline of the thesis	9
2	Frame of reference	11
2.1	Precision Agriculture	11
2.2	Global navigation satellite systems	12
2.2.1	Improving GNSS accuracy.....	13
2.2.2	Error sources	14
2.3	Autonomous agricultural vehicles.....	15
2.4	Wheel odometry	16
2.5	Visual Odometry	17
2.5.1	Related work	17
2.5.2	Camera models	19
2.5.3	Visual odometry pipeline	22
2.6	Visual simultaneously localization and mapping	24
2.6.1	Related work.....	24
3	Research approach	27
3.1	Philosophy of science.....	27
3.2	Methodology	28
3.3	Rigour and relevance	29
3.4	Scientific paradigm	29
4	Summary of included papers	31
4.1	Paper I: Algorithms for visual odometry in outdoor field environment	31
4.2	Paper II: Stereo visual odometry for mobile robots on uneven terrain .	33
4.3	Paper III:A vision-guided mobile robot for precision agriculture	35

4.4	Paper IV: Row-detection on an agricultural field using omnidirectional camera	37
4.5	Paper V: Ego-motion estimation by an agricultural field robot using visual odometry	38
4.6	Paper VI: Localization of agricultural field robots using natural landmark and visual SLAM	40
5	Discussion	43
5.1	Future work	45
6	Conclusions	47
	References	51

LIST OF FIGURES

1.1	Manual weeding at a vegetable plant. Photo: Linnéa Jarefjäll. Courtesy of Mariestadstidningen (Moberg, 2013).	2
1.2	An autonomous agricultural robot used for weed-control in sugar beet fields, courtesy of Åstrand (2005).....	3
1.3	Relationship between the publications included in this thesis. The symbol with one camera represents work performed on a monocular camera. The double camera symbol represents a stereo camera, while the camera with a circular lens represents an omnidirectional camera.	5
2.1	Principle of a differential GNSS using a mobile internet link between rover unit and base station.....	14
2.2	Setup of two encoder wheels to estimate position of a mobile robot. ...	16
2.3	Model of a monocular perspective camera.	19
2.4	Model of a stereo camera.....	20
2.5	Calibration of the forward-facing stereo cameras using a checkerboard. Images of the checkerboard are captured from different directions and distances. The checkerboard used contains 9 x 7 squares each 0.1 x 0.1 m. Each grid represents the checkerboard position for one frame. The numbers in the figure represent frame numbers.	21
2.6	Visual odometry pipeline as presented in D. Scaramuzza and Fraundorfer (2011).....	22
3.1	An instantiation of the information systems research framework. Adapted from Hevner et al. (2004).....	28
4.1	Sample images from the data sets collected by the mobile platform used for experiments in Paper I. The scale is used as ground truth in the experiment.	32
4.2	Matching scheme for stereo images used in Paper II. The height and distance travelled are estimated using the measured disparities. k is the frame number and d is the disparity.....	33
4.3	Experimental setup used in Paper II.....	34
4.4	A selected result of the Paper II experiment. Plot of dx disparity representing the height of ten test runs with respect to x-position. The ground surface is uneven soil with sugar beet.....	34

- 4.5 Robot model on the field as used in Paper III. The distance to the row (s) and heading angle (α) is measured by the row-following system. The steering angle (β) is controlled so the robot tracks the rows. The visual odometry system measures the travelled distance along the row. 35
- 4.6 Example image from the row detection system used in Paper III. a) shows the captured color image, b) shows the binary image after color filtering and segmentation, c) shows the Hough space plotted in α -s-plane. Each maximum (white fields) represents one row. 36
- 4.7 Camera model used in Paper IV: Projection through the lens on to the sensor plane by a unit sphere model of the omnidirectional camera 37
- 4.8 Result from Paper IV showing detected rows on a potato field captured with a fisheye camera. Blue lines show rows detected as lines, and red lines show rows detected using a Hough transform. 38
- 4.9 The field experiment used in Paper V and Paper VI. The plot from Google Earth shows the robot path on the field for three test runs. Map data: Google, Lantmäteriet, Metria. The left-hand track in white represents the Field 1 experiment and the right-hand track in white the Field 2 experiment. The red track is the path of the Field 4 experiment. 39
- 4.10 Resulting plot from using the ORB-SLAM stereo algorithm with forward-facing cameras on the Field 3 sequence; black line = ground truth, blue line = SLAM result, open circle = start position, solid circle = end position. This is the best result for all tests. There are, however, no loop closures and error accumulation still remains. 41
- 5.1 Perspective of individual blades of grass, where the trackable intersection in image plane does not represent camera motion 44

LIST OF TABLES

2.1	Required accuracy for agricultural application (Wolfgang Lechner and Baumann, 2000)	12
4.1	Result of experiments on grass in Paper I	32
4.2	Result from Paper III where the robot drove 10 meters autonomously along the row. The reported distance of the visual odometry system is compared to the position measured using a ruler.....	36
4.3	Results from Paper V. Visual odometry on an oat and fodder field with a sub-sample of the data set.....	40

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

The use of autonomous robots to carry out everyday human tasks has become more common in recent years. Robotic mowers and vacuum cleaners are popular because they relieve their owners of tasks that are sometimes both boring and onerous. Yet researchers regard these tasks as relatively simple because they involve reliable sensors and robust algorithms. The environment in which these robots operate has a defined boundary that can be robustly detected by the sensors (Hicks and Hall, 2000; Fiorini and Prassler, 2000). Robotic cleaners use walls to define the area, while most robotic mowers use a buried wire. The robot moves randomly or in pre-programmed pattern within the area as it cleans or mows.

Another area that could benefit from using autonomous robots is crop production (Blackmore et al., 2008; Bechar and Vigneault, 2016). Agricultural fields also have defined boundaries and there are specific tasks to be carried out. The main differences between this environment and the environment in which domestic robots operate are the size of the fields and the types of tasks to be carried out. Field operations in crop production include ploughing, harrowing, sowing, fertilizing, weed control, and harvesting. The size of the field requires a large robot that is strong enough to perform the task within a reasonable time, which increases the demand for robustness and reliability. Sometimes the time slot in which an action can be executed is very small. For example, harvesting has to be done at a specific time to optimize harvest quality and to avoid bad weather that would make harvesting impossible. Farmers have to work almost around the clock at harvest time to maximize the quantity and quality of the crop. Even though this task requires only a tractor driver, it creates an unequal work load with a huge peak during harvesting.

The long-term goal is to develop autonomous robots that can carry out agricultural tasks like weed control in an economical and ecologically sustainable way. There are many publications showing the technical feasibility of using agricultural robots for different crops, tasks and abilities (Bechar and Vigneault, 2016). Yet there are still technical challenges when it comes to enabling the robots to cope with the difficult, highly unstructured, and dynamic agricultural environment. The biggest technical challenge in developing autonomous agricultural robots relates to reliability and robustness. Robustness is defined here as the ability to work in various environmental



Figure 1.1: Manual weeding at a vegetable plant. Photo: Linnéa Jarefjäll. Courtesy of Mariestadstidningen (Moberg, 2013).

conditions, such as different weather conditions. An agricultural robot must also be able to handle dirt, moisture, and different temperature and light conditions. Rain makes fields slippery and creates challenges for locomotion and position estimation. An autonomous robot should also be able to detect and avoid obstacles.

From an economic perspective, the main obstacle to the development and use of commercial autonomous agricultural vehicles is the seasonality of agriculture, which makes it hard to achieve high levels of utilization (Bechar and Vigneault, 2016).

One task that is particularly suited to automation is weed control in organic farms. This is a very labor-intensive task, for no pesticides are allowed on such farms and so weed control has to be done by hand. Figure 1.1 shows a woman doing this (Moberg, 2013) at a vegetable plant. She is lying on a special trailer on which up to six persons can lie prone while picking weeds. The trailer is pulled along by a tractor moving at 0.6 km/h, and each person is responsible for one row.

On cornfields there is especially one weed that needs to be picked by hand, the wild oat (*Avena Fatua*). This tricky weed competes for resources with crop plants, and the presence of even a few in a wheat or oat field will lower the yield (Länsstyrelsen, 2016). Farmers in Sweden are obliged by law (Svensk Författningsamling 1970:299) to control wild oats, and there are joint efforts to do so by farmers, the government and various organizations. Hand picking of wild oats is only one of several actions against the weed. During the summer, there are teams available ready to walk the fields to remove wild oat by hand.

An example of an agricultural robot that was developed for automatic weed control is shown in Figure 1.2 (Åstrand and Baerveldt, 2005; Åstrand, 2005). A forward-facing camera is used to detect the row structure of a sugar beet field, and the robot is programmed to follow these rows. A downward-facing camera is used to detect

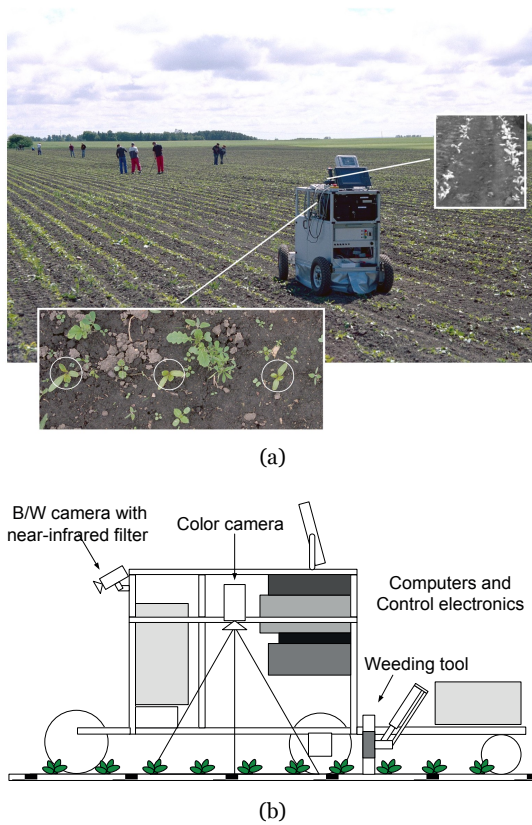


Figure 1.2: An autonomous agricultural robot used for weed-control in sugar beet fields, courtesy of Åstrand (2005).

plants and to distinguish weeds from the crop. Approximately one meter behind the camera is a rotating hoe that mechanically removes weeds. Mechanical weeding creates a lot of dirt, and the distance between the hoe and the camera is necessary to avoid lens occlusion. An encoder wheel is used to measure the weeding tool's position relative to the plant identification camera. The precision of the tool is determined by its mechanical performance, the precision of the plant position, and the robot's position and heading on the field. One drawback of the encoder wheel is that it suffers from wheel slippage, which makes the estimation of the tool position inaccurate.

The main focus of this thesis is on vision techniques to estimate the position of a tool relative to a plant identification camera. In other words, the focus is on localization in the local frame. The advantages of a vision system are that it is a non-contact sensor and can be used for other tasks such as crop identification, obstacle detection, and monitoring. The main localization sensor for autonomous agricultural vehicles is a high accuracy GPS receiver, which provides position estimates in 3 degrees of freedom (DoF). It can provide 6-DoF in combination with other sensors such as an inertial navigation system (INS). Visual odometry, a vision-based system for measuring relative motion, can provide full pose (position and heading angle) of the robot in 6-DoF. It is also possible to build maps by using techniques for simultaneous localization and mapping (SLAM). Combining GPS and a vision system could

potentially increase both the reliability and robustness of the system.

1.2 AIM AND OBJECTIVES

The aim of this project is to increase knowledge about visual localization in an outdoor field environment to enable automation of field operations by autonomous robots.

This aim is broken down into the following objectives:

- Identify a suitable algorithm for localization using visual odometry.
- Evaluate the impact of different design options in visual odometry.
- Develop a navigation system based on row-following and visual odometry.
- Evaluate different algorithms for visual SLAM on real field data.

In order to identify the most effective algorithm, suitable data sets had to be acquired or generated. A mobile robot had to be developed in order to collect data in a real agricultural field. In addition, different simulators had to be developed to complement the real field tests to target specific factors that are impractical to test in the field.

1.3 RESEARCH QUESTIONS

The research questions are structured with one main research question that is split up into three sub-groups. Each sub-group contains a set of questions. The main research question that this thesis sets out to answer is this:

How and in what way can cameras be used for localization of an agricultural field robot?

The three sets of sub-questions arising from the main question are as follows:

VISUAL ODOMETRY

- How can visual odometry improve the relative localization accuracy of an agricultural robot?
- What algorithms should be used?
- What is a suitable sensor setup?
- How do the different design options in the visual odometry pipeline affect the localization result?

ROW-FOLLOWING

- How can a row-following system be used in combination with visual odometry for navigation?
- How can the robustness of existing row-following systems for agricultural robots be improved?

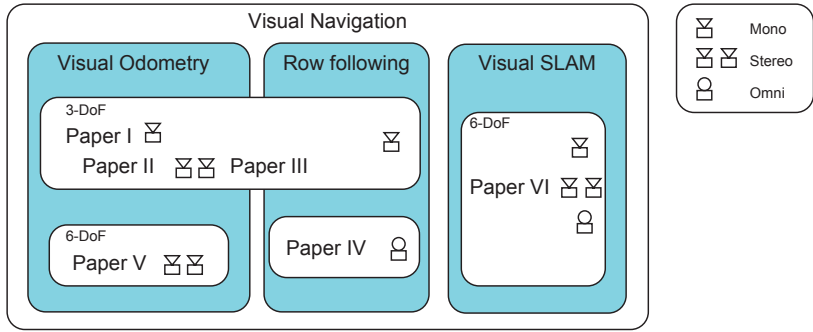


Figure 1.3: Relationship between the publications included in this thesis. The symbol with one camera represents work performed on a monocular camera. The double camera symbol represents a stereo camera, while the camera with a circular lens represents an omnidirectional camera.

VISUAL-SLAM

- How suitable are visual SLAM algorithms for navigation of agricultural robots?
- What algorithms should be used?
- What is a suitable sensor setup?

1.4 RELATIONSHIP BETWEEN ARTICLES

This work started with investigating how translation could be estimated using downward-facing cameras. The aim was to use it on an agricultural robot instead of wheel odometry in cases where a downward camera was already present for crop identification. Visual odometry based on four different algorithms was evaluated on various ground textures using a monocular camera. The work was extended to stereo cameras. A visual odometry algorithm was developed that provides position estimation in 3-DoF and that works on uneven terrain. The work on visual odometry was finalized by evaluating two algorithms to provide 6-DoF pose. The effect on the odometry result of the different steps in the visual pipeline was evaluated on simulated data. Finally, the algorithms were tested on real field data. This work answers the research questions regarding visual odometry and led to three publications (Ericson and Åstrand, 2007; Ericson and Åstrand, 2008; Ericson and Åstrand, 2016).

The 3-DoF visual odometry system was implemented on a mobile robot to run in real time. It was combined with a row-following system that autonomously controls the robot to drive along the rows. An improved row detection algorithm was developed that combines detection of solid lines with lines of single plants. This work answers the two research questions regarding row-following and led to two publications (Ericson and Åstrand, 2009; Ericson and Åstrand, 2010).

Finally visual SLAM algorithms were evaluated on real field data, which answers the last research question regarding visual SLAM. This work led to one publication (Ericson and Åstrand, 2017). Figure 1.3 shows the relation between the research questions and the publications included in this thesis.

1.5 MAIN CONTRIBUTIONS OF THE INCLUDED ARTICLES

The author of this thesis is the main author of all the included articles. He designed and conducted all experiments, including developing research platforms and simulations, and wrote the papers. The co-author of the papers contributed guidance and feedback on the work and on the reports. The long gap between Paper IV and Paper V was the result of parental leave and of the author being head of his division. The older paper is still highly relevant; the same or similar results would be obtained if the experiments were repeated.

PAPER I

Stefan Ericson and Björn Åstrand (2007). “Algorithms for Visual Odometry in Outdoor Field Environment”. In: *Proceedings of 13th IASTED International Conference Robotics and Applications*. Würzburg, Germany, pp. 287–292

In this paper presented at the 13th IASTED International Conference on Robotics and Applications in Würzburg, Germany, visual odometry algorithms are evaluated for use on an autonomous agricultural weeding robot. The robot is equipped with a weed detection camera and a mechanical weeding tool. An encoder wheel keeps track of the weeding tools position relative to the camera. As the wheel suffers from wheel slip, crop plants are sometimes removed instead of weeds. This specific problem can be solved by using the weed detection camera with a visual odometry algorithm to estimate the position of the weeding tool. Two feature-based methods and two pixel-based (dense) methods were evaluated to estimate 2D position from consecutive frames captured by a monocular camera. The feature-based method extracts and matches salient points, in this case using Harris corners (Harris and Stephens, 1988) and a scale-invariant feature transform (SIFT) (Lowe, 2004). The pixel-based approaches use most of the pixels in the image for matching. The two methods evaluated are template matching using normalized cross correlation (Jähne, 2002) and the Lukas-Kanade method (Lucas and Kanade, 1981). The ground is assumed to be flat and the distance to the ground is known. The methods were tested on four different surfaces: carpet, asphalt, grass, and soil. This paper’s contribution is as follows:

- A comparison of four algorithms for visual odometry, two feature-based and two pixel-based.
- An analysis of the effect of different ground textures. Four textures were tested: carpet, asphalt, grass and soil.
- An evaluation of the planar ground assumption.

PAPER II

Stefan Ericson and Björn Åstrand (2008). “Stereo Visual Odometry for Mobile Robots on Uneven Terrain”. In: *IAENG Special Edition of the World Congress on Engineering and Computer Science, Advances in Electrical and Electronics Engineering*, pp. 150–157

In this paper, presented at the World Congress on Engineering and Computer Science 2008 conference in San Francisco, USA, we presented a visual odometry system for agricultural field robots that is not sensitive to uneven terrain. A stereo camera system is mounted perpendicular to the ground. Height and distance travelled

are calculated using normalized cross correlation. Laboratory experiments were designed in which flower boxes containing representative surfaces were placed in a metal-working lathe. The cameras were mounted on the carriage, which can be positioned manually with 0.1 mm accuracy. Images were captured every 10 mm over a 700 mm traverse. The tests were performed on eight different surfaces representing real world situations. This paper's contribution is as follows:

- A scheme for template matching to make height-invariant position estimates.
- A validation gate to select the most likely match even when there is a poor texture and aperture problems occur.
- Experiments on real sugar beet with accurate ground truth in the laboratory.

PAPER III

Stefan Ericson and Björn Åstrand (2009). "A vision-guided mobile robot for precision agriculture". In: *Proceedings of 7th European Conference on Precision Agriculture*. Wageningen, Netherland, pp. 623–630

This paper was presented at the 7th European Conference on Precision Agriculture in Wageningen, the Netherlands. We developed a mobile robot controlled only by vision sensors. The system consists of a row-following system and a visual odometry system. The row-following system captures images from a forward-facing monocular camera on the robot and the crop rows are extracted using a Hough transform. Both distance to the rows and heading angle are provided and both are used to control the steering. The visual odometry system uses two cameras in a stereo setup facing downwards to the ground. This system measures the travelled distance by measuring the ground movement and compensates for height variations.

This paper demonstrates the feasibility of a mobile robot controlled only by vision sensors.

PAPER IV

Stefan Ericson and Björn Åstrand (2010). "Row-detection on an agricultural field using omnidirectional camera". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010. IROS 2010*, pp. 4982–4987

This paper presented at the IEEE/RJS International Conference on Intelligent Robots and Systems (IROS), 2010, Taipei, Taiwan, describes a method of detecting parallel rows on an agricultural field using an omnidirectional camera. The method works both on cameras with a fisheye lens and cameras with a catadioptric lens. A combination of an edge-based method and a Hough transform method is used to both detect individual rows and to find the vanishing point of several parallel rows. The method was evaluated on synthetic images generated by a simulator which uses calibration data from real lenses. Scenes with several rows were produced, where each plant is positioned with a specified error. Experiments were performed on these synthetic images and on real field images.

This paper's contribution is as follows:

- A method to detect parallel rows on agricultural fields using an omnidirectional camera.

- A method that can detect both rows with individual plants and solid lines.
- A method that can be used both for cameras with catadioptric lenses and for cameras with fisheye lenses.

PAPER V

Stefan Ericson and Björn Åstrand (2016). “Ego-motion estimation by an agricultural field robot using visual odometry”. Submitted to Biosystems Engineering

This paper, submitted to Biosystems Engineering, analyzes two visual odometry systems for use in an agricultural field environment. The effects of various design parameters and camera setups were evaluated by simulation. Four real field experiments were conducted using a mobile robot operating in an agricultural field. The robot was controlled to travel in a regular back-and-forth pattern with headland turns. The experimental runs were 1.8 to 3.1 km long and consisted of 32,000–63,000 frames. The best results were obtained using high resolution images captured using forward-facing stereo cameras mounted with a large baseline. The algorithm must be able to reduce error accumulation by adapting the frame rate to minimize error. The results also illustrate the difficulties of estimating roll and pitch using a downward-facing camera. The best results for full 6-DoF position estimation were obtained on a 1.8-km run using 6680 frames captured from the forward-facing cameras. The translation error (x, y, z) is 3.76% and the rotational error (i.e., roll, pitch, and yaw) is $0.0482 \text{ deg m}^{-1}$.

This paper’s contribution is as follows:

- An analysis of the effect of design options on visual odometry results.
- A head-to-head comparison of two visual odometry algorithms in an agricultural setting.
- Evaluation of the algorithms on agricultural data.

PAPER VI

Stefan Ericson and Björn Åstrand (2017). “Localization of agricultural field robots using natural landmark and vision SLAM”. Submitted to Sensors

In this paper, submitted to the MDPI journal Sensors, we evaluated two visual SLAM algorithms, the feature-based ORB-SLAM and the dense LSD-SLAM, on a challenging agricultural data set. The data set was acquired using a mobile robot equipped with a set of cameras and driven on an open agricultural field in a regular back-and-forth pattern. The resulting data set consists of four sequences where the path length varies between 1.8 and 3.1 km. Both monocular and stereo versions of the algorithms were evaluated. The LSD-SLAM method was also applied to omnidirectional images.

This paper’s contribution is as follows:

- A comparison study of two visual localization and mapping methods in an agricultural setting.
- Real data from four different data sets using a mobile robot on fodder and oat fields.

1.6 OUTLINE OF THE THESIS

Chapter 2 provides the frame of reference for this thesis, presenting the background and key technologies from both agricultural and computer vision perspectives. Chapter 3 explains the research approach and method, The included papers are summarized in chapter 4, and the findings are discussed in chapter 5, followed by conclusions in chapter 6.

CHAPTER 2

FRAME OF REFERENCE

This chapter begins with a short introduction to precision agriculture (Section 2.1). Then the technology and limitations of global satellite navigation systems are explained in Section 2.2. Section 2.3 gives an overview of autonomous robots developed for use in agriculture. A description of the technology of visual odometry along with related work is presented in Section 2.5, while Section 2.6 describes the methods used in visual SLAM.

2.1 PRECISION AGRICULTURE

A farmer who cultivates crops needs to make many decisions based on many variables. The weather is probably the variable with the greatest impact on the crop, but it is also the hardest to predict over an entire growing season. When is the best time to sow? How much fertilizer is needed? How to protect against weeds? When is the best time to harvest? A farmer builds up knowledge and makes decisions based on experience. When a farmer has to work on a field, the entire field is treated the same way unless the farmer knows some reason not to do so.

Precision agriculture is the research field focusing on the use of information technology and sensors to manage spatial and temporal variations in agricultural production in order to improve crop production and environmental quality (Pierce and Nowak, 1999). The goal is to allow decisions to be based on site-specific measurements so that the correct treatment can be applied with precision at the right time and position.

The most important sensor in precision agriculture is the global navigation satellite system (GNSS), which provides a measurement of both position and time. This enables other measurements such as variabilities in topology, soil type, and soil moisture content to be accurately positioned on a map (Zhang, M. Wang, and N. Wang, 2002). Measurements can then be combined with other types of data such as aerial or satellite images and weather forecasts to provide a better decision support system to the farmer (Herring, 2001).

One of the first applications, and most typical examples, of precision agriculture is fertilizing. A sensor measures the light reflectance of the crops to determine how much fertilizer is needed (Link, Panitzki, and Reusch, 2002). Then the right amount

of fertilizer is applied directly to the crops. In this way each part of the field is treated based on its needs, which is optimal for both the crop and the environment.

The research area of precision agriculture includes sensors, measurement methods, mapping, machinery, robotics, management, and decision systems (Stafford, 2013). It also includes agricultural robots that are used for data acquisition and for operations such as sowing, weeding, and harvesting. The precision of these robots determines what types of task can be executed. High precision systems can perform crop scale operations, while low precision systems can operate on only a section of a field. Tasks like precision spraying or mechanical weed control require high precision of measurement relative to the crop.

The introduction of real-time kinematic (RTK) GNSS made it possible to monitor and guide agricultural vehicles with an accuracy of a few centimeters (Bell, 2000; Gan-Mor, Clark, and Upchurch, 2007). A typical application is to install RTK GNSS on a tractor, where it can show the tractor's path on a screen. This means that the driver does not have to look back over his or her shoulder to verify that the implement is following the correct path, which improves the driver's posture. Automatic steering can also be added, allowing autonomous control of the tractor. In both cases the tracks can be defined in advance so that row overlap can be minimized, saving labor, fertilizer and fuel. Even though the equipment is relatively expensive, its cost can be recouped in a few months, depending on total field size (*John Deere Autotrac* 2016).

Different tasks require different position accuracy. Table 2.1 shows different tasks and the required resolution (Wolfgang Lechner and Baumann, 2000).

Goal	Application	Accuracy
Recognition of fields	Recording working hours Recording machine hours	± 20 m
Recognition of parts of the field	Optimal local distribution yield mapping	± 1 m
Guidance of machines	Connection drives	± 0.05 m
Guidance of working tools	Working on the plants	± 0.01 m

Table 2.1: Required accuracy for agricultural application (Wolfgang Lechner and Baumann, 2000)

These requirements can be compared with those obtained from different types of GNSS receiver. A low-cost non-differential GPS with an accuracy of 3–30 meter can be used to recognize fields. An augmented GPS with 3-meter accuracy can be used to recognize parts of a field, while a high performance RTK GNSS with an accuracy of 0.02 meter is required for machine guidance and guidance of working tools. A detailed explanation of the different GNSS technologies is given in the next section.

2.2 GLOBAL NAVIGATION SATELLITE SYSTEMS

A global navigation satellite system (GNSS) provides an absolute measurement of 3D position and an accurate time to a user with a passive receiver (Krüger, Springer, and W. Lechner, 1994; Wolfgang Lechner and Baumann, 2000; *NAVSTAR GPS User Equipment Introduction* 1996). The first and most used system is the US military system called the Global Positioning System (GPS). Similar systems have been developed by other countries, including the Russian system GLONASS, the European system Galileo, and the Chinese system COMPASS. In addition to these global

systems, there are a few regional systems.

The NAVSTAR GPS (Navigation System with Time and Ranging – Global Positioning System) was developed for the US Air Force (*NAVSTAR GPS User Equipment Introduction* 1996) and is maintained by them. It was developed primarily for the military, but one of the two signals was made available for civilian use. This signal was intentionally scrambled by a system called selective availability (SA) to reduce the accuracy to about 300 m. On 1 May 2000 the US government turned off selective availability, giving civilian users higher accuracy. The goal was to stimulate civilian use, and today there are more civilian than military users of GPS. However, the USA can still turn on SA without any warning globally or in regions where there are conflicts.

The satellite system is based on positioning by triangulation (*NAVSTAR GPS User Equipment Introduction* 1996). A satellite sends a unique code to a receiver, including when and from where the signal was sent. Thus the signal's time of flight can be measured and the distance to the satellite can be calculated. Signals from four satellites are required to be able to compute X, Y, and Z positions. The fourth satellite is required to compute the time error between the cheap receiver clock and the satellite's accurate atomic clock. The position of all four satellites is also sent to the receiver, which means that the receiver works completely passively. The accuracy of the obtained position is 3–30 m. The term GPS is commonly used instead of GNSS to describe a satellite-based position system. There are receivers today capable of using several systems at the same time.

2.2.1 IMPROVING GNSS ACCURACY

The basic principle for enhancing the accuracy of satellite navigation systems is to measure or model the errors and correct for them (Wolfgang Lechner and Baumann, 2000). One error source is signal delays in the atmosphere, usually in the ionosphere. This error is modelled and corrections are sent by geostationary satellites at the same frequency as the GNSS. Thus the corrections can be received by a regular receiver and the user does not need other hardware. Since the correction satellites are regional, each part of the world has its own augmentation system. The European system is called EGNOS (European Geostationary Navigation Overlay Service). The Wide Area Augmentation System (WAAS) covers North America, and the Japanese MSAS (Multifunctional Transport Satellite-based Augmentation System) is an augmentation system covering Asia and Pacific regions. The error is reduced from 20 m to 3 m by using these corrections. This method has the drawback that it may be hard to receive correction signals from the satellite at higher latitudes, since each satellite has a fixed position close to the horizon.

Even better accuracy can be obtained if the reference station is closer to the user. Such a system can be built with two GNSS receivers, where one is at a known fixed reference point. Both receivers measure the position and the reference receiver calculates an error based on the difference between the measured position and the actual position. This information is sent to the other receiver on a special communication link. This technique is called differential GNSS (Wolfgang Lechner and Baumann, 2000). To use differential GNSS the receivers have to be able to transmit and receive this additional correction. The position is still based on the pseudorandom code transmitted from the satellites. The typical error is in the range of 0.5–3 m.

More sophisticated techniques can increase the position accuracy of differential GNSS. Instead of comparing the position obtained from the transmitted code, the phases of the incoming carrier signal can be compared. This increases the accu-

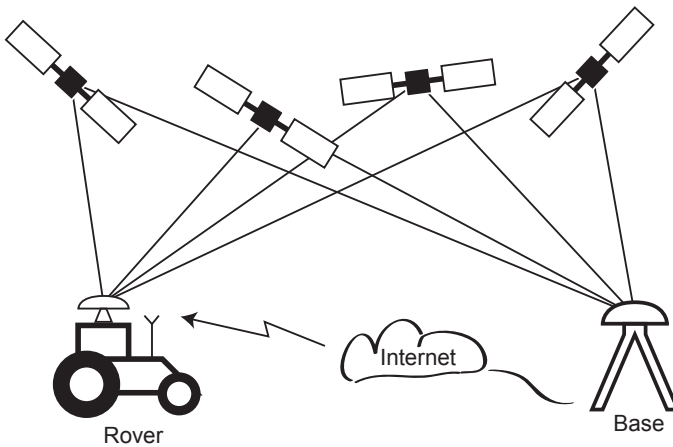


Figure 2.1: Principle of a differential GNSS using a mobile internet link between rover unit and base station.

racy significantly; the typical error is 0.01 m horizontally and 0.02 vertically. Once the carrier phase is locked on the satellites, the position can be estimated with high frequency, which makes it possible to use it in real-time control systems for vehicles. The name real-time kinematic GPS (RTK-GPS) comes from this ability. The technique is used on autopilots on tractors, but it is also used for static position measurement in construction and geographic information systems (GIS).

The correction data can also come from a network of RTK receivers that provides RTK correction data. In this case only one RTK receiver is required. In Sweden there is a system called Network RTK (Lantmäteriet, nodate). Similar networks exist or are under construction in other countries. The network consists of reference stations with RTK receivers at calibrated positions. Correction data is calculated and transmitted to the user via the cellular phone network. A subscription is required for this service and there are fees. The same accuracy is obtained as if the correction came from its own local base station. The biggest advantages of using this system are that no reference unit is required for the user and that RTK correction can be obtained seamlessly.

2.2.2 ERROR SOURCES

The basic principle of GNSS is that the receiver should have line-of-sight (free path) access to at least four satellites (*NAVSTAR GPS User Equipment Introduction 1996*). If fewer than four satellites are visible to the receiver, no position will be obtained at all, which is referred to as a drop out. This problem can be solved either by moving to another position where reception is better or by waiting for the satellites to move into a better position.

A multipath signal, which occurs when the signal is reflected off the ground, or more commonly off buildings, violates the free path to satellite requirement. The signal path is then longer, and the triangulation of the position will be wrong. This effect can be observed when a position jumps from one spot to another as soon as a multipath signal is captured and used for position estimation. This error cannot

be compensated for, and also occurs with RTK GNSS. The best protection against this error is to mount the antenna on a metal plate to shield it from signals reflected from the ground. It is more difficult to shield against reflections from buildings.

As mentioned previously, delays in the atmosphere also decrease the accuracy of the measured distance to the satellite. These types of error can be reduced by using an augmentation system. But the accuracy also depends on where in the sky the satellites are. The principle of triangulation requires the satellite positions to be distributed across the sky. The theoretical ideal would be for the satellites to be orthogonal to each other, for example, one directly overhead, one at the southern horizon, and one at the eastern horizon, but that is rare in practice. Two or more may be orthogonal, but it is difficult to have a clear path to satellites just above the horizon. That is why the vertical error is usually larger than the horizontal error.

2.3 AUTONOMOUS AGRICULTURAL VEHICLES

One of the first fully autonomous agricultural robots was Demeter (Pilarski et al., 2002), a modified harvester guided by a regular GPS and a vision system. The advantage of combining GPS and vision is their complementary features. GPS provides position with no long-term drift, while the vision system can be used without a map and is capable of detecting obstacles. They also have different failure modes, which is good for sensor fusion. When the GPS provides a measurement with low accuracy, the vision system can take over and, together with wheel encoders, provide a more accurate measurement. The vision system is used for line following, end of crop detection, and obstacle detection. However, one major issue for agricultural field robots and vision is the shadows cast by the robot on sunny days (Pilarski et al., 2002). If these shadows fall into the camera's field of view, some areas will have a much lower intensity than others. A method using spectral power distribution of RGB images can be used to classify points as being either in shadow or in sun.

Although Demeter demonstrated the feasibility of autonomous agricultural robots a long time ago, there are, to the author's knowledge, no commercial autonomous agricultural robots available yet. However, there are several research platforms such as Hortibot (Jørgensen et al., 2007), BoniRob (Ruckelshausen et al., 2009), Gantry (Jiang et al., 2014), Thorvald (Grimstad et al., 2015). All these robots are designed to have omnidirectional drive by having four combined steering and driving wheels, one at each corner of the robot. The BoniRob (Ruckelshausen et al., 2009) differs slightly from the other robots in that the wheels are mounted on legs so that both the height and the wheelbase of the robot can be adjusted. All these robots demonstrate navigation skills, and some even demonstrate the use of an implement. There are also concept vehicles such as Case IH Agriculture's autonomous tractor (Atherton, 2016). Companies such as Trimble¹ and John Deere² offer autonomous drive for tractors. A more comprehensive review of research in robotics for field operations can be found in (Bechar and Vigneault, 2016). It shows that there are many publications on the technical feasibility of using agricultural robots for different crops, tasks and abilities. In this thesis, vision localization technologies are evaluated to further increase knowledge about their abilities, performance, and limitations.

¹Trimble Agriculture - <http://www.trimble.com/agriculture/>

²John Deere - <http://www.deere.com/>

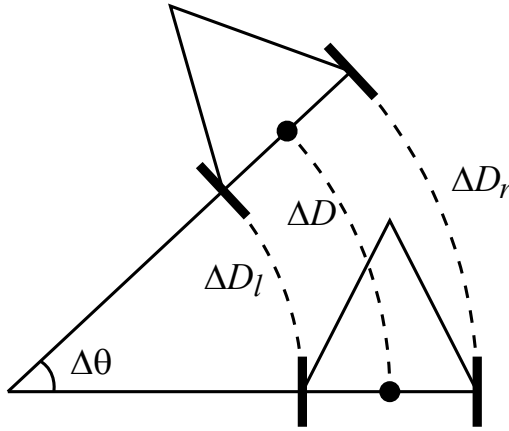


Figure 2.2: Setup of two encoder wheels to estimate position of a mobile robot.

2.4 WHEEL ODOMETRY

Wheel odometry is a simple and common method of estimating a robot's position. This is a relative measurement, determined by counting wheel rotations. A rotary encoder placed on the axis of a wheel can provide an accurate measurement of the wheel's rotation. To convert this rotation measurement to a travelled distance, two assumptions have to be made. The first is that the wheel diameter is constant and known, and the second is that the wheel does not slip on the ground. Given these assumptions, the travelled distance of a single wheel can be calculated according to eq. (2.1).

$$\Delta D = W \cdot \alpha \quad (2.1)$$

where ΔD - is the travelled distance, W - wheel diameter, α - wheel rotation in radians.

If two of these wheels are mounted on the left and right side of a mobile robot according to Figure 2.2, then the position of the robot can be estimated in 3-DoF. Let D_r denote the travelled distance of the right wheel and D_l the distance of the left wheel. Then the position (x, y) and heading θ of the center point between the wheels can be calculated according to eq. (2.2) and eq. (2.3), given that the wheel baseline L is known (C. Wang, 1988).

$$\begin{aligned} \Delta D_n &= (\Delta D_r + \Delta D_l)/2 \\ \Delta \theta_n &= (\Delta D_r - \Delta D_l)/L \end{aligned} \quad (2.2)$$

$$\begin{aligned} x_n &= x_{n-1} + \Delta D_n \cos \left(\theta_{n-1} + \frac{\Delta \theta_n}{2} \right) \\ y_n &= y_{n-1} + \Delta D_n \sin \left(\theta_{n-1} + \frac{\Delta \theta_n}{2} \right) \\ \theta_n &= \theta_{n-1} + \Delta \theta_n \end{aligned} \quad (2.3)$$

where n is the sample number.

The advantage of wheel odometry is that it is a very simple way of estimating relative position. The main drawbacks are that error accumulates over time and that the assumption of fixed wheel diameter and no wheel slippage fails. This is especially true in the case of agricultural robots, which commonly use air-filled rubber wheels and negotiate slippery fields.

2.5 VISUAL ODOMETRY

Position estimation using consecutive images from an on-board camera is usually referred to as visual odometry. The term covers a wide range of meaning, from the robot's full pose estimation in a global frame to position estimation along one direction in the local frame. The term "camera ego-motion estimation" is also commonly used. The principle is to capture images of the surroundings and then compare consecutive frames to find differences (D. Scaramuzza and Fraundorfer, 2011; Fraundorfer and D. Scaramuzza, 2012). Assuming that most of the image is occupied by static objects, the relative motion can be derived from this information. In contrast to wheel odometry, visual odometry does not suffer from errors caused by wheel slip or varying wheel diameter. However, the problem of error accumulation still remains.

The relative pose change of a camera between two consecutive frames is modelled as a rigid motion transform. This transform is shown in eq. (2.4) (D. Scaramuzza and Fraundorfer, 2011).

$$dT_r = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.4)$$

where $\mathbf{R} \in SO(3)$ is the rotation matrix and $\mathbf{T} \in \mathfrak{R}^{3 \times 1}$ is the relative translation. The cumulative pose C_k at frame k can be obtained using eq. (2.5):

$$C_k = C_{k-1} dT_r \quad (2.5)$$

The goal is to find dT_r and C_k for each frame. It should be mentioned that the transformation matrix is an overdetermined system and can be rewritten using Euler angles as $\mathbf{v} = (x, y, z, roll, pitch, yaw)$.

Visual odometry differs from visual landmark recognition in that visual odometry does not store data for further use. Hence the error of a visual odometry system is cumulative and long-term stability requires either mapping ability or fusion with other sensors that provide global position estimations. In landmark recognition, frames or features are stored and are used as reference for new frames. This method requires larger storage space and an algorithm that compares the current frame to the frames in the database. This method will be further presented in Section 2.6.

2.5.1 RELATED WORK

The term "visual odometry" was introduced by Nistér, Naroditsky, and Bergen (2004), pioneers in the field. They demonstrated visual odometry for both monocular and stereo cameras by extracting feature points, in this case Harris corners (Harris and Stephens, 1988), which are matched for correspondence and used for motion estimation. An efficient solution of the five-point problem is presented in (Nistér, 2005), which makes it possible to only use five image points to fully estimate the relative pose of the camera from two consecutive images. The feature matcher produces many false matches, and to handle these outliers random sample consensus

(RANSAC) is used (Nistér, 2005). A mobile robot with forward-facing cameras was used to acquire data. The best result showed a distance error of 1.1% measured at the end position of a 186 m long path.

Another pioneer was Matthies, who derived error models that can be used in stereo navigation (L. Matthies and Shafer, 1987). This is important when combining visual odometry with other sensors or with a motion model. His work was implemented on the Mars rovers only as an extra feature, but it turned out that visual odometry played an important role due to heavy wheel slip while driving on Martian sand (Maimone, Cheng, and Larry Matthies, 2007). The importance of detecting and keeping track of errors was found when evaluating visual odometry for autonomous ground vehicles (Howard, 2008). Howard stated that a single mismatch in the motion estimation can introduce a large error, despite thousands of correct estimations. Many algorithms have been presented since then, but the major differences lie in what features to extract and how to remove outliers (Kitt, Geiger, and Lategahn, 2010; Geiger, Ziegler, and Stiller, 2011; Cvišić and Petrović, 2015).

In order to evaluate and make a fair comparison between different visual odometry algorithms a publicly available data set has been published online, the KITTI Vision Benchmark Suite³ (Geiger, Lenz, and Urtasun, 2012). It consists of a total of 21 image sequences captured from a car with forward-facing cameras. The cameras are mounted on the roof of the car together with an inertial navigation system to provide ground truth. The inertial navigation system consists of a RTK-GPS and an inertial measurement unit (IMU) so that the full 6-DoF pose is obtained as reference. The sequences are mainly from urban environments, with some exceptions where the car is driven on a highway.

The ground truth for the first eleven sequences is public, but the remaining ten are kept secret and are used as a benchmark. The results obtained with different algorithms can be submitted to the site and evaluated against the ground truth. The error is calculated for all sub-sequences of length 100 m, 200 m, ..., 800 m and summarized over the entire path. The final error is presented as translation measured in percent of travelled distance and rotation as drift in degrees per meter. The highest ranked result in June 2016 for visual odometry using a stereo camera was *SOFT* (Cvišić and Petrović, 2015), which achieved a translation error of 0.88% and rotation error of 0.0022 deg m⁻¹. In November 2016 there was a new leading algorithm *SOFT2* from an anonymous submission, which slightly improved the translation error to 0.81%. This accuracy is achieved by carefully selecting the features so only stable features that are easy to track are used for motion estimation.

The feature extractor used in the *SOFT* algorithm is available in a C++ visual odometry library called *Libviso* (Geiger, Ziegler, and Stiller, 2011). It extracts several different types of corners/blobs by finding minimum and maximum points in images filtered using 5 x 5 templates. A blob and checkerboard pattern is used in the template. Descriptors of these points are created using gradients in a neighborhood, which are used when matching points to each other.

The *Libviso* library will also be used in agricultural applications (*Kameras als Augen des Jät-Roboters* 2015), but no results have been published yet. In fact there are few publications related to the use of visual odometry in agriculture. Jiang et al. (2014) presents a robot called the *Gantry* used on a soybean field. The robot is three meters high and shaped like a table with one wheel at each corner. On the top of the robot there is downward-facing stereo camera used for visual odometry. Reported results on a 2.5 km test in which the *Gantry* was driven in a back-and-forth pattern on the field show an error of 0.2% in translation only.

³The KITTI Vision Benchmark Suite <http://www.cvlibs.net/datasets/kitti/index.php>

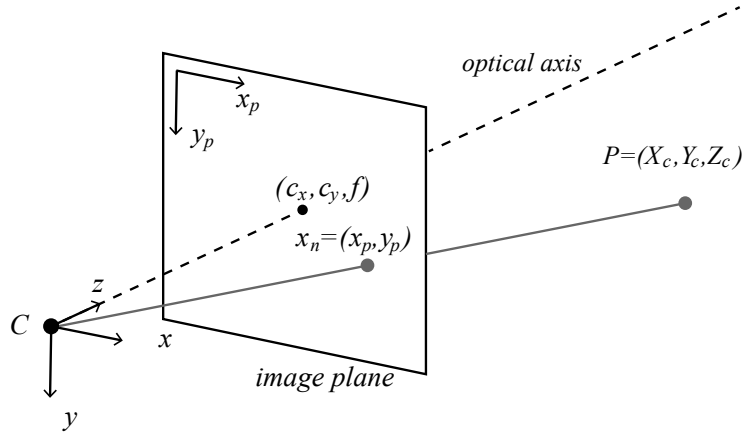


Figure 2.3: Model of a monocular perspective camera.

A major problem in research into agricultural applications of visual odometry is that there are no agricultural data sets similar to the KITTI Benchmark Suite, nor any similar comparative rankings. Instead each author designs their own experiments and reports the results, which makes it hard to compare algorithms.

2.5.2 CAMERA MODELS

A camera model is required to map 3D objects in the camera's field of view onto the 2D image plane, and vice-versa. A perspective camera can be modelled using a pinhole camera model, see Figure 2.3. Then a point in 3D-space $P = (X_c, Y_c, Z_c)^T$ can be projected onto the image plane using eq. (2.6)-(2.10) (*Camera calibration With OpenCV* 2016; Bouguet, 2008).

Let $(x, y)^T$ be the normalized image projection:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} \quad (2.6)$$

The camera lens creates distortion that can be modelled as radial distortion. In this model three parameters are used (k_1, k_2, k_3) . If the lens is not perfectly parallel to the image plane, tangential distortion is created. This can be modelled using two parameters (p_1, p_2) . Let $r^2 = x^2 + y^2$. Then the image point corrected for distortion can be written as:

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_4 r^6) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 3p_2 xy \end{bmatrix} \quad (2.7)$$

The pixel coordinates (x_p, y_p) can then be calculated according to:

$$\begin{cases} x_p = f_x(x_d + \alpha_c \cdot y_d) + c_x \\ y_p = f_y y_d + c_y \end{cases} \quad (2.8)$$

where (f_x, f_y) is the focal length measured in pixels, and (c_x, c_y) is the optical center expressed in pixel coordinates. α_c is the skew coefficient between the x and y pixel axis, which is zero for orthogonal axes.

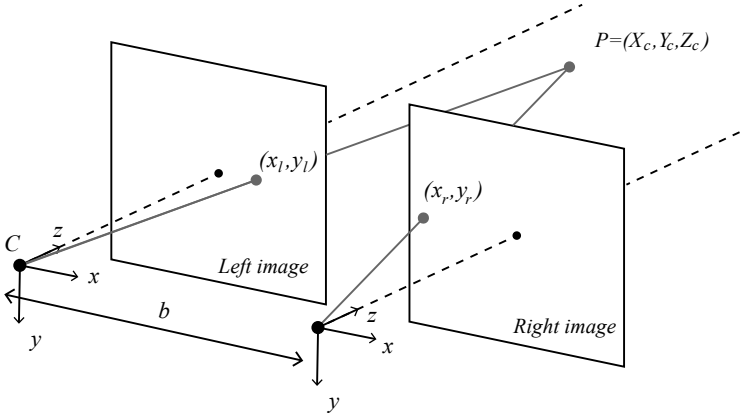


Figure 2.4: Model of a stereo camera.

The focal length can be set to $f = f_x = f_y$ if the pixels are square, and the same conversion factor between pixels and meters can be used in the x and y directions. Equation (2.8) can be rewritten using the camera matrix:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \quad (2.9)$$

where the camera matrix is defined as:

$$K = \begin{bmatrix} f_x & \alpha_c \cdot f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

The camera has to be calibrated in order to determine its parameters. The procedure for calibrating the camera is to capture a set of images of a checkerboard from different distances and angles. The corners of the squares in the checkerboard are identified and matched to each other between the images. Radial and tangential distortion makes the checkerboard look bent. The distortion parameters, along with the camera's focal length and principal point (camera intrinsic parameters), are obtained from the calibration.

The limitation of using a monocular camera is that depth information is lost. The depth can be restored if two cameras identify the point from different positions. A stereo camera can be modelled as two monocular cameras where the relative camera position is represented by a rigid motion transform, i.e., the extrinsic parameters. A common setup is to let the cameras' optical axes be parallel at the distance b , which is the baseline, see Figure 2.4.

Let the image point in the left image expressed in image coordinates be (x_l, y_l) and in the image from the right camera be (x_r, y_r) . Then the disparity can be calculated as $d_x = x_l - x_r$. The position of the corresponding point in 3D space (X_c, Y_c, Z_c) can then be calculated according to eq. (2.11).

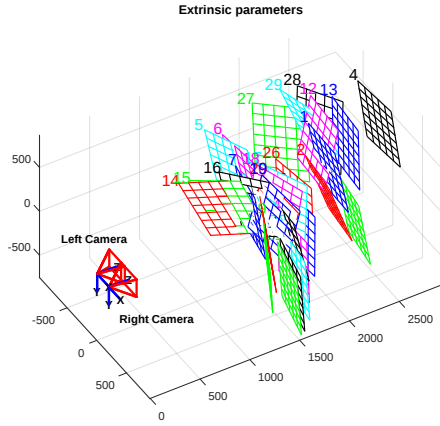


Figure 2.5: Calibration of the forward-facing stereo cameras using a checkerboard. Images of the checkerboard are captured from different directions and distances. The checkerboard used contains 9×7 squares each 0.1×0.1 m. Each grid represents the checkerboard position for one frame. The numbers in the figure represent frame numbers.

$$\begin{aligned} X_c &= \frac{b \cdot x}{d_x} \\ Y_c &= \frac{b \cdot y}{d_x} \\ Z_c &= \frac{b \cdot f}{d_x} \end{aligned} \quad (2.11)$$

where b is the baseline and f is the focal length.

Usually the camera is not perfectly aligned as described in the model, but by knowing the rotation matrix and a translation vector describing their relative position (extrinsic camera parameters), the images can be rectified. The rectification aligns the epipolar lines with the image's x-axis. That means that for each pixel in the left image, the corresponding point in the right image is located on the same row. That simplifies the model and can also be used to reject false matches. In practice both undistortion and stereo rectification can be processed at the same time by remapping all pixel points using a lookup table. This is a fast method, but it assumes that the camera parameters are fixed.

The calibration of a stereo camera is similar to the monocular case. Images of a checkerboard pattern have to be captured using a stereo camera that gives one set of images from the left camera and one set from the right camera. Then each camera is calibrated to find its individual intrinsic parameters. The final step estimates the extrinsic parameters of the cameras by matching the pattern between the camera views. The intrinsic parameters are optimized as well, to minimize the overall calibration error. Figure 2.5 shows the checkerboard's location relative to the forward-facing stereo camera.

There are toolboxes that can be used for calibration, and in this thesis three different toolboxes have been used. The calibration of perspective cameras as described above used the toolbox described in (Bouquet, 2008). The omnidirectional camera calibration uses two different models (Mei and Rives, 2007; D. Scaramuzza, Mar-

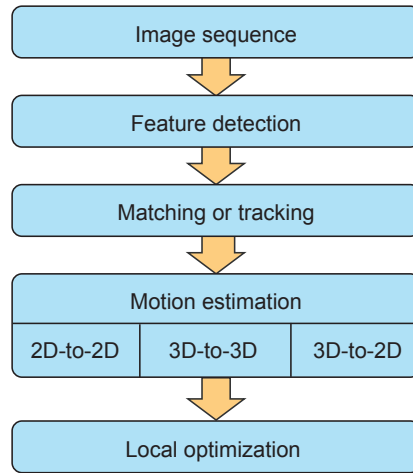


Figure 2.6: Visual odometry pipeline as presented in D. Scaramuzza and Fraundorfer (2011).

tinelli, and R. Siegwart, 2006). Thus two different toolboxes were used to calibrate the cameras. However, the procedure using a checkerboard is the same.

2.5.3 VISUAL ODOMETRY PIPELINE

The steps in the visual odometry pipeline (Figure 2.6) are presented in a comprehensive two-part tutorial on visual odometry (D. Scaramuzza and Fraundorfer, 2011; Fraundorfer and D. Scaramuzza, 2012). This section will give a brief introduction and explain the main concepts, but for a more detailed explanation the reader is referred to the original publications.

IMAGE SEQUENCE

The algorithm is fed image sequences captured by cameras on the robot. The placement of the camera is one design option: it should be set up to capture images in which the major part is a static environment. Most algorithms work on grayscale images, and a common resolution is 640×480 pixels. If the sequence consists of color images, these have to be converted to grayscale. It may, however, be preferable to acquire grayscale images directly from the camera since such images may be of higher quality than converted images, depending on the camera technology. Another design option is whether to choose a monocular or stereo setup. As previously discussed, a monocular setup can only estimate position up to a scale factor, while a calibrated stereo setup does not have that limitation and can provide a full 6-DoF pose estimation. Finally, overlap between images has to be ensured for the operation range of the robot. The parameters that affect the overlap are the camera's frame rate and the maximum velocity of the robot.

FEATURE DETECTION AND MATCHING

There are two approaches to finding correspondence between images, a feature-based approach and a dense approach. The feature-based approach is to extract

salient features in the image and then tag these points with a descriptor. The descriptor has to be invariant to position, but invariance to rotation and scale extends the range of translation and rotation between consecutive frames that can be estimated. The descriptors are matched to each other in order to identify corresponding points between stereo cameras and consecutive frames. The descriptors should be as unique as possible so each match is unambiguous. The match is usually performed back-and-forth to further ensure that the match is correct. This is referred to as a left-right consistency check in the case of stereo images. In the stereo visual odometry case where there are two stereo pairs that should be matched, it is possible to make a quad loop check. Then a point should find a match in all four images, and should then be matched back to the starting point to complete the loop. This method is available in *Libviso* (Geiger, Ziegler, and Stiller, 2011). A rule of thumb is to adapt the detector so that 1000 features are found in an 640×480 pixel image (Fraundorfer and D. Scaramuzza, 2012). A few hundred of these will be matched to each other, but many of these matches will be wrong.

The other approach is to use a dense or area-based approach to directly find correspondences between all image pixels. This method relies on a filtering technique. A template, usually the neighborhood of a fixed point in the reference image, is used to find similarities within the other image (Nourani-Vatani and Borges, 2011). For real-time performance, the search area is limited to a minimum, given the probability of finding a correct match. However, the search area still has to be large enough to capture all expected motion changes. This method is usually used with downward-facing cameras (Zaman, 2007; P. Hansen et al., 2011; Peter Hansen et al., 2011; Nourani-Vatani and Borges, 2011) and where the motion can be represented by translation. There are algorithms that can extract rotation and affine transforms, but these are computationally more expensive (Zitova and Flusser, 2003). An advantage of dense methods is that all information in the entire image is used, in contrast to feature-based methods that use only information around a feature point.

MOTION ESTIMATION

Motion is estimated from the matches between two consecutive frames. There are three different methods depending on how the motion is estimated, 2D-2D, 3D-2D, and 3D-3D (Fraundorfer and D. Scaramuzza, 2012). In the 2D-2D method the relation between corresponding pixel points is estimated using the so-called essential matrix, which is then used to estimate the motion. This method is usually associated with monocular cameras. In the 3D-2D method a 3D point cloud is calculated using stereo triangulation. The point cloud's projection onto the second frame is compared with the corresponding image points (2D) found in the second frame. The translation and rotation that minimizes the reprojection error gives the motion of the camera. Finally there is a 3D-3D approach that calculates two point clouds in 3D. The differences correspond to the camera motion. The latter two methods are commonly used together with stereo cameras for the simplicity of finding the point's depth. A fundamental requirement of visual odometry is that the points come from a static environment. Points violating this requirement should be treated as outliers and should be removed. Most algorithms require that the majority of the found points belong to the static environment representing a common motion (that of the camera). An efficient way of handling data with a large number of outliers is to use random sample consensus (RANSAC) (Fischler and Bolles, 1981; Hartley and Zisserman, 2004).

LOCAL OPTIMIZATION

The final step in the visual odometry pipeline is local optimization, which is a technique for minimizing the error over several consecutive frames. This can be done either by using pose-graphs where the nodes are represented by camera poses and the edges by rigid motion transforms (Olson, Leonard, and Teller, 2006). A cost function describing the sum of all translation and rotation errors is introduced and minimized. This is the same technique used in visual SLAM, but the distinction is that it should be local optimization that is effective only over a set of consecutive frames. Another successful approach is bundle adjustment (Triggs et al., 2000), which uses the same approach with pose-graphs but optimizes the position of the 3D feature points. This builds a map that makes the visual odometry equivalent to visual SLAM. The technique of using a windowed or local bundle adjustment where the position is optimized over a number of consecutive frames is presented as the last step in the visual odometry pipeline (Fraundorfer and D. Scaramuzza, 2012). D. Scaramuzza and Fraundorfer (2011) argues that the distinction between visual odometry and visual SLAM is that in visual odometry the aim is to estimate the local trajectory as well as possible, i.e. to achieve local consistency, while visual SLAM aims to find global consistency. In this thesis the distinction between visual odometry and visual SLAM is the presence of an optimization step that stores and uses old images to reduce drift.

2.6 VISUAL SIMULTANEOUSLY LOCALIZATION AND MAPPING

Given a robot with the ability to perceive its surroundings, it is relatively easy to build a map based on that perception as the position of the robot is known. If, on the other hand, there is already a map, it is relatively easy to estimate the robot's position. Simultaneous localization and mapping (SLAM) tries to solve these two tasks simultaneously. The special case where a camera is used for perception is referred to as visual SLAM. SLAM terminology uses the word "features" to describe *landmarks* that can be used in the map. To avoid confusion is the word landmark referred to SLAM mapping point in this thesis and feature to an image point. However, features found in the image can be used as landmarks in a SLAM algorithm (Stachniss, 2013).

2.6.1 RELATED WORK

The algorithms used for solving the general SLAM problem can be categorized into three paradigms based on their solution method: Kalman filter approaches, particle filter approaches, and graph-based approaches (Roland Siegwart, Nourbakhsh, and Davide Scaramuzza, 2011; Stachniss, 2013).

In the Kalman filter approach the robot's pose and the position of all landmarks are assembled together into an extended Kalman filter (EKF). This enables both the localization (robot's pose) and the map (position of the landmarks) to be solved simultaneously. The size of the covariance matrix will increase as the robot explores new areas and additional landmarks are added to the map. This leads to a system that is computationally intensive. It is most suitable for systems with up to one thousand landmarks. The Kalman filter requires that the noise can be modelled as Gaussian and that both bearing and distance to the landmark can be measured simultaneously. This is not the case in monocular visual SLAM where only the bearing to the landmarks is given. This problem can be solved by initiating the map with

position estimates of all landmarks.

In particle filter SLAM the poses are represented by samples (particles) from the probability distribution. Weights are assigned to each particle based on their importance and then resampled. Thus the particle filter approach is not limited to Gaussian probability distributions but can handle any type of distribution. Another advantage is that, in contrast to the Kalman filter approach, it can handle multiple hypotheses. However, a drawback is that for some distributions a large number of particles are required to be fully representative. A low number of particles can lead to overconfidence in false hypotheses, the so-called particle depletion phenomenon. Another drawback is that non-linear systems with higher degrees of freedom, such as SLAM, have to be linearized. FastSLAM is an effective SLAM implementation based on the particle filter (Michael Montemerlo et al., 2002; M. Montemerlo et al., 2003).

The graph-based approach follows the strategy previously described in Section 2.5.3 where the nodes in the pose-graph represent poses of the robot and landmark positions, while the edges represent their relative positions. This system can be seen as a network of springs, where the solution is obtained by minimizing the energy in the net (Roland Siegwart, Nourbakhsh, and Davide Scaramuzza, 2011). The advantages of this approach are that it can handle outliers well and is computationally less expensive. The computational complexity is $\mathcal{O}(N)$ compared to EKF-SLAM that has complexity $\mathcal{O}(N^2)$, which makes it able to handle maps with more than a thousand landmarks.

The specific problem of using visual SLAM is that the position of the selected landmarks has to be measurable. This requires a stereo camera setup but, as mentioned before, this can be solved by initializing landmark positions in the map allowing monocular SLAM to be used. As with visual odometry, both dense and feature-based approaches can be used with SLAM. An example of a feature-based approach is the parallel tracking and mapping (PTAM) algorithm that was developed for artificial reality where virtual 3D objects can be included in the camera view with the correct projection even though the camera is moving (Klein and Murray, 2007). The implementation consists of two threads, where one thread estimates the position of the camera while the other is building a map. Even though PTAM shares the problem of simultaneously performing localization and mapping, their different applications limit the use of PTAM in large-scale environments. It lacks functionality such as loop closure and has difficulties handling occlusions. It also suffers from low invariance to different viewpoints for relocalization (Mur-Artal, Montiel, and Tardós, 2015).

ORB-SLAM (Mur-Artal, Montiel, and Tardós, 2015) is a state-of-the-art algorithm that was specifically developed for visual SLAM. This is a feature-based approach is similar to PTAM, but ORB-SLAM use ORB features (Rublee et al., 2011) rather than PTAM's FAST corners. It includes both place recognition and loop closure detection by using a visual-word concept (Yang et al., 2007). This is implemented in a third thread that is run in parallel with the tracking and mapping threads. The mapping is similar to PTAM and uses selected keyframes and a graph-based SLAM approach to find global consistency. The ORB-SLAM method was also evaluated on the KITTI benchmark data set for visual odometry and the reported error was 1.15% for position and $0.0027 \text{ deg m}^{-1}$ on rotation. It should be noted that the algorithm does fail on one sequence from a highway where there are too few trackable features. That particular sequence is the one that is most similar to the data set of an open agricultural field.

An alternative to extracting feature is to use a dense approach that uses all the information in the image. That overcomes the shortcoming of feature-based methods

that can use only information that conforms to the feature type (Jakob Engel and Daniel Cremers, 2014). Large-scale direct monocular SLAM (LSD-SLAM) (Jakob Engel and Daniel Cremers, 2014) is one implementation of a dense approach. Instead of features, filter-based estimation is used to create semi-dense depth maps. A graph-based SLAM approach is used where the vertices represents keyframes and the edges a 3D similarity transform. The stereo implementation of the algorithm (J. Engel, Stückler, and D. Cremers, 2015) has been evaluated on the KITTI benchmark suite, and the reported result is 1.20% translation error and $0.0033 \text{ deg m}^{-1}$ rotational error.

The use of SLAM in agriculture is commonly associated with applications where the plants or trees have visible stems and are positioned in rows (Cheein et al., 2011; Hiremath et al., 2014; Lepej and Rakun, 2016). The robots move between or around these rows, and a 2D laser scanner is used to detect the stems. In (Cheein et al., 2011) the laser scanner is combined with a monocular camera to detect stems in an olive grove. The SLAM algorithm used an extended information filter (EIF), which is classified as a Kalman filter approach. They showed that the robustness increased by fusing data from two different sensors. Their result shows an error of less than 0.5 m when navigating around a 70×70 m field. Another SLAM approach is used in (Hiremath et al., 2014), which used a particle filter to perform the SLAM. They demonstrated the method on a maize field using only a 2D laser scanner sensor to detect stems. The reported RMS error of the position is 0.04 m and 2.4 deg.

A different solution to the SLAM problem is presented in (Lepej and Rakun, 2016). They also detected stems using a 2D laser scanner, but in an apple orchard and a vineyard. The scans were then transformed into images. Image registration techniques were applied to find correspondence between scans. The error of the reported 3-DoF position is $0.16\% \pm 0.1\%$ on a path with a total length of 252 m.

CHAPTER 3

RESEARCH APPROACH

After a short introduction to the philosophy of science in Section 3.1, Section 3.2 deals with the scientific method and more specifically with the methodology adopted in the research presented in this thesis. The rigour and relevance are discussed in Section 3.3 and finally, the paradigms in computer science are described in Section 3.4.

3.1 PHILOSOPHY OF SCIENCE

The philosophy of science is the study of science from a philosophical perspective (Klemke et al., 1998). It deals with questions about what science is and is not, how it works, and the logic required to build scientific knowledge. The standard philosophical view of natural science is empiricism, which argues that all knowledge is built on observational and empirical evidence (Robson, 2002). This positivistic approach assumes the existence of an objective reality shared by all observers. It also assumes that all phenomena are controlled by natural laws, discoverable by systematic observation and experimentation (Fyfe, 2005). Singular statements then become universal statements in the form of hypotheses or theories. However, the problem with inductive inferences is exemplified by the white swan hypothesis (Popper, 2002). No matter how many white swans we observe, it does not justify the conclusion that all swans are white.

A post-positivistic view of research argues that research evidence is always imperfect and liable to error (Robson, 2002). Therefore all knowledge comes with no warranty since it is based solely on the best evidence we have at the time. Research is the process of making claims that are abandoned or refined in the light of evidence. Post-positivists believe that there is an external reality separate from and independent of our perception of it. This view is in line with the philosophical theory of realism. In contrast to empiricism, rationalism argues that reason and not experience is the primary source of knowledge. Knowledge is a social and historical construct, and all problems can be solved using nothing but reason. This approach is common in the social science and is based on deductive reasoning.

The philosophical theory of instrumentalism was introduced by Popper (2002) and Dewey, Boydston, and Gouinlock (2008). It argues that scientific theories are in-

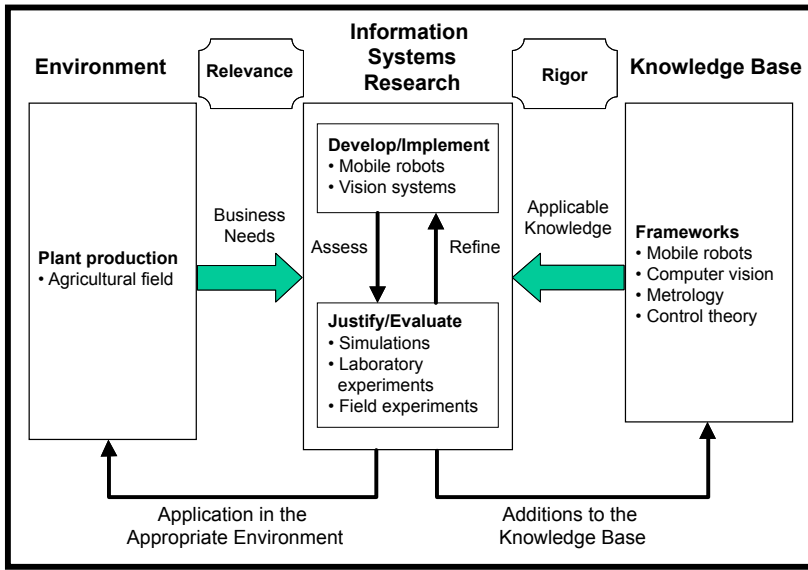


Figure 3.1: An instantiation of the information systems research framework. Adapted from Hevner et al. (2004)

struments and that their effectiveness in explaining and predicting phenomena determines their worth. The instrumentalists do not believe that theories are true or false, or that they correctly represent reality; instead theories solve practical problems. In contrast to scientific realism, instrumentalism does not try to use theories to describe the natural world (Britannica, 2016).

In my own work I hold to a positivistic philosophy, in which experiments are performed and compared to a ground truth. The ground truth in this case is a related measure that provides better accuracy with known error. The goal of my analysis is to find explanations or models that accurately predict behavior. In this way more knowledge is gained about a specific phenomenon.

3.2 METHODOLOGY

The choice of research method is crucial for the conclusions that can be drawn from observations and experiments. It affects what you can say about the cause and factors influencing the phenomenon. The research approach used in this thesis is Design and Build (Oates, 2006; Hevner et al., 2004). Research is defined as the creation of new knowledge, using an appropriate process, to the satisfaction of the user of the research. The Design and Build method focuses on the development of IT products and artefacts to gain knowledge (Oates, 2006). The design is considered as both a process and a product (the artefact), which is refined iteratively based on the experience from the process and evaluation of the model. This combination requires strategies associated with both the behavioral science and design-science paradigms, which is summarized in a information systems research framework (Hevner et al., 2004). An instantiation of this framework adapted to the research question in this thesis is found in Figure 3.1.

One artefact developed in this work is the visual odometry system, which is used as an instantiation of visual localization. The process included design of both hardware

and software using an engineering design strategy. Each stage in the evolution of the artefact has resulted in an improvement in both the steps in the design process and the resulting artefact. The Design and Create method also accommodates user perspectives on the artefact, which may require the use of methods from the behavioral science. However, in the present study, no such evaluation was undertaken. Instead experiments verifying the functionality of the artefacts were performed and the technical performance of the system was evaluated using a quantitative method. The new knowledge gained from each artefact is used to develop a new version of the artefact with improved performance.

3.3 RIGOUR AND RELEVANCE

Research in information science should be characterized by rigor and relevance. “Rigor” requires that research be performed in a rational fashion based on a systematic approach, and that proper methods are used to ensure the result is valid. “Relevance” is defined as pertinence to practitioners (Oates, 2006). The practitioners in this thesis are other developers of localization and navigation systems for agricultural robots. “Relevance” here does not directly address farmers, since the scope of this work does not include full scale autonomous agricultural robots.

The work presented in this thesis follows a systematic method in which each step in the design process and implementation is verified and the performance evaluated. Validity is taken into account when designing experiments. For example, consider the evaluation of the impact of different design options in a visual odometry system, such as a camera baseline. If two separate field experiments were done to evaluate two baselines, it could be argued that there are so many variables that might be different (like the weather, light, path), that the results would have low internal validity. To avoid this problem, the experiment would have to be designed to evaluate different baselines simultaneously, requiring a stereo camera setup for each. This would be very costly and impractical, which is why a simulator was developed to generate the required data sets while changing only the baseline.

The external validity is lower in the field experiments conducted on real agricultural fields, since these are just samples of the world at a specific time. However, these experiments provides valuable knowledge about the complexity of the problem and the performance of current technology in that environment.

3.4 SCIENTIFIC PARADIGM

Computer science is a wide research field that incorporates many different areas. According to (Eden, 2007) there are three paradigms, the rationalist paradigm, the technocratic paradigm and scientific paradigm. The rationalist paradigm treats computer science as a branch of mathematics which use deductive reasoning to seek a priori knowledge about programs. This paradigm is common among theoretical computer scientists. The technocratic paradigm treats programs as data and posteriori knowledge is build by empirical tests. This approach is the most dominating in computer science and is practioned by many software engineers. The last paradigm is the scientific paradigm which defines computer science as natural science. Programs are treated as mental processes and priori and posteriori knowledge is build upon deduction and experiments. This paradigm is prevalent in artificial intelligence.

The work in this thesis belongs to the technocratic paradigm, where engineering methods are dominating. The different algorithms’ performance are evaluated by

experiments. The main strategy when developing algorithms are to use traditional mathematical modelling, which if successful would give the best solution. Another option would have been to use artificial intelligence to let the algorithm itself learn and find best solution based on some score function. A drawback of this approach is that there is no guarantee that the best solution is found, since the result can end up in a local optima. And if the best solution is found it would provide the same result as the solution obtained by mathematical analysis. However, artificial intelligent has been successful in computer vision to identify objects, and research on plant classification would probably benefit from using self-learning methods.

CHAPTER 4

SUMMARY OF INCLUDED PAPERS

4.1 PAPER I: ALGORITHMS FOR VISUAL ODOMETRY IN OUT-DOOR FIELD ENVIRONMENT

Stefan Ericson and Björn Åstrand (2007). “Algorithms for Visual Odometry in Outdoor Field Environment”. In: *Proceedings of 13th IASTED International Conference Robotics and Applications*. Würzburg, Germany, pp. 287–292

In this paper presented at the 13th IASTED International Conference on Robotics and Applications in Würzburg, Germany, visual odometry algorithms were evaluated for use on an autonomous agricultural weeding robot equipped with a weed detection camera and a mechanical weeding tool. An encoder wheel kept track of the weeding tool’s position relative to the camera. However, the wheel suffered from wheel slip which resulted in removing crop plants instead of weed. A solution to this specific problem was to use the weed detection camera with a visual odometry algorithm to estimate the position of the weeding tool. Two types of visual odometry algorithms were evaluated: feature-based using Harris corners and scale-invariant feature transform (SIFT) features, and pixel-based or dense algorithms using normalized cross correlation (NCC) and the Lukas-Kanade method.

The Harris corner method extracts Harris corners that are matched for correspondence using a local neighborhood. The SIFT approach uses scale and rotation invariant descriptors that are matched between feature points. The normalized cross correlation method uses a template image, which is a small region of one image, that is matched for similarity to another image. The most similar area is selected as a match. The Lukas-Kanade method uses gradients in the image and is an iterative method that allows correspondence with sub-pixel accuracy. Once a match is found, the motion is estimated by calculating travelled distance from the image disparity. It is assumed that the ground is flat and the distance to the ground is known. The final 2D position is obtained by accumulating all motion contributions.

The methods were evaluated on data sets collected using a mobile platform with a downward-facing monocular camera. The platform was moved by hand in straight lines, moving forward 1 meter and then stopping, after which the platform was moved back to the starting point. A scale visible in the images was used as ground truth, but this part of the image was not used by the visual odometry algorithms.

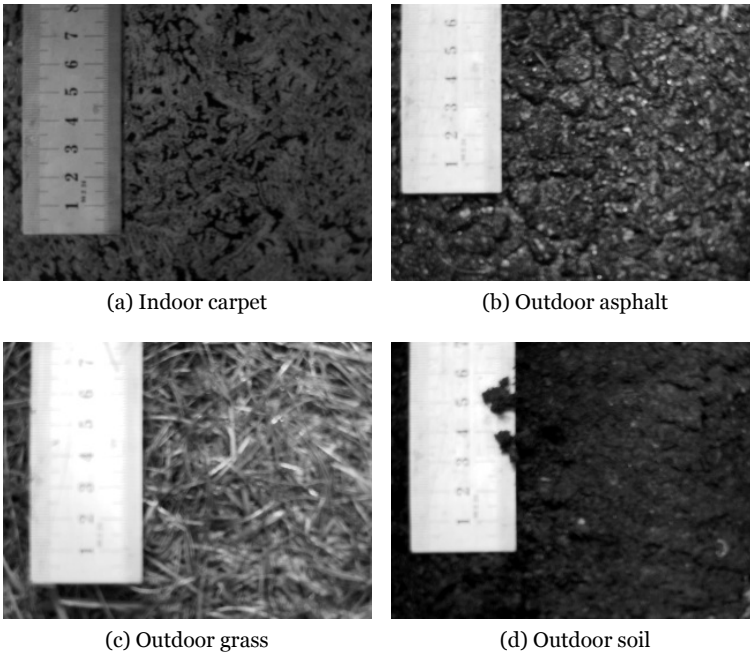


Figure 4.1: Sample images from the data sets collected by the mobile platform used for experiments in Paper I. The scale is used as ground truth in the experiment.

Data were acquired from four different ground surfaces: carpet, asphalt, grass, and soil. Figure 4.1 shows sample images of the different surfaces. Table 4.1 presents the results showing the accuracy of the position estimation for one of the surfaces.

Test Method	Error 1 m	Error back & forth	Calculation time
Harris corner	1.1%	3.1%	323 ms
SIFT	-1.2%	0.3%	586 ms
NCC	-1.7%	0.6%	144 ms
Lucas-Kanade	-1.4%	0.2%	302 ms

Table 4.1: Result of experiments on grass in Paper I

The results show that the feature-based approach using SIFT gave a good result, but was computationally more expensive than all other methods. Using Harris corners gave many matched features but many of them were wrong, which led to a poor result. It should be mentioned that no powerful outlier rejection was used, just a majority vote. The optical flow method using Lucas-Kanade method did perform well, but could only accommodate small changes between images even though a pyramid approach were use. The template matching method using normalized cross correlation provided the best overall result. The full results (not included here) showed that the main difference between the textures was the ground flatness. Some tests were performed by driving back-and-forth over the surface, which cancels the height error, leaving only the error from the visual odometry. The results also showed that the assumption of flat ground does not hold, since there is a large variation when

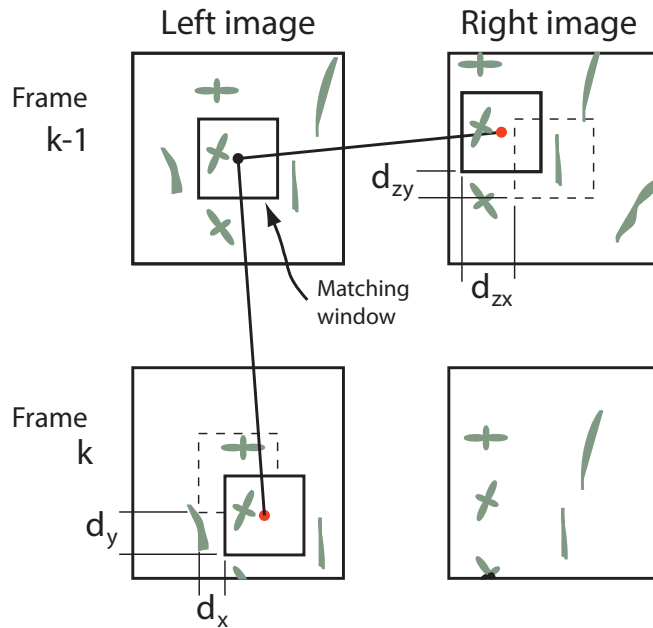


Figure 4.2: Matching scheme for stereo images used in Paper II. The height and distance travelled are estimated using the measured disparities. k is the frame number and d is the disparity.

the robot drives on grass.

4.2 PAPER II: STEREO VISUAL ODOMETRY FOR MOBILE ROBOTS ON UNEVEN TERRAIN

Stefan Ericson and Björn Åstrand (2008). “Stereo Visual Odometry for Mobile Robots on Uneven Terrain”. In: *IAENG Special Edition of the World Congress on Engineering and Computer Science, Advances in Electrical and Electronics Engineering*, pp. 150–157

This paper, presented at the World Congress on Engineering and Computer Science 2008 conference in San Francisco, presents a visual odometry algorithm for agricultural field robots that is not sensitive to uneven terrain. A stereo camera system was mounted perpendicular to the ground. Height and travelled distance were estimated using normalized cross correlation. The matching scheme is shown in Figure 4.2. The same template was used for both stereo matching and position estimation, leaving only one calibration parameter remaining, the camera baseline. Each match resulted in a similarity image in which the pixel values vary between zero for no correspondence and one for a perfect match. A good match provides a clear peak where the maximum is 1. This point was selected in the validation gate in such a way that the best position estimate was obtained.

Laboratory experiments were designed with flower boxes containing representative surfaces in a metal-working lathe. The cameras were mounted on the carriage, which could be positioned manually with 0.1 mm accuracy. Images were captured



Figure 4.3: Experimental setup used in Paper II

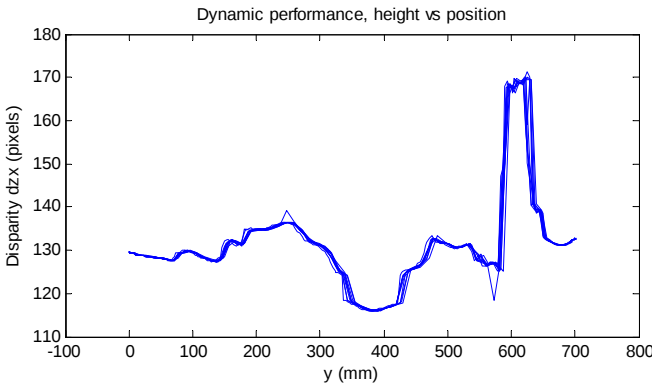


Figure 4.4: A selected result of the Paper II experiment. Plot of dx disparity representing the height of ten test runs with respect to x-position. The ground surface is uneven soil with sugar beet.

every 10 mm over a 700 mm length. The tests were performed on eight different surfaces, including soil, sugar beet, and grass, representing real world situations. Figure 4.3 shows the test setup with the blue cameras mounted on the carriage. The flower box containing grass is on the floor below.

The result shows that the system can provide position estimates with an error of 0.59% of travelled distance, even though the height may vary by up to 30% of the camera clearance. Figure 4.4 shows a plot of the disparity representing the height with respect to the estimated position for ten test runs on one surface. This result shows that the height estimation was stable over ten different test runs with respect to the position.

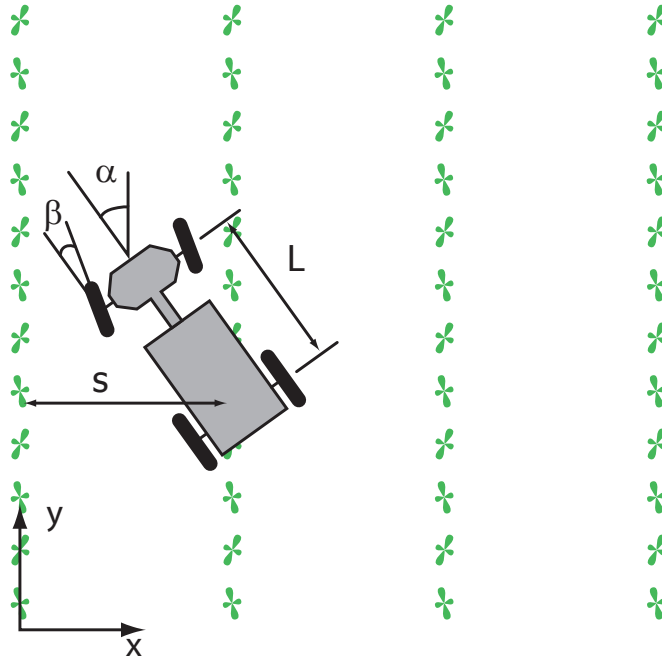


Figure 4.5: Robot model on the field as used in Paper III. The distance to the row (s) and heading angle (α) is measured by the row-following system. The steering angle (β) is controlled so the robot tracks the rows. The visual odometry system measures the travelled distance along the row.

4.3 PAPER III: A VISION-GUIDED MOBILE ROBOT FOR PRECISION AGRICULTURE

Stefan Ericson and Björn Åstrand (2009). “A vision-guided mobile robot for precision agriculture”. In: *Proceedings of 7th European Conference on Precision Agriculture*. Wageningen, Netherland, pp. 623–630

This paper, presented at the 7th European Conference on Precision Agriculture in Wageningen, the Netherlands, describes a mobile robot controlled only by vision sensors. The system consists of a row-following system and the visual odometry system developed in Paper II. The row-following system captures images from a forward-facing monocular camera on the robot and the crop rows are extracted using a Hough transform. Two rows are used to estimate both distance to the rows and relative heading angle, which are used to control the steering so that the robot follow the rows. The visual odometry system was implemented for real-time performance and the cameras were mounted in a box behind the robot’s back wheel. Figure 4.5 shows the robot model, showing distance (s) and heading angle (α) to the row measured by the row-following system, the robot length L , and the steering angle (β).

The results show an error of about 2% of travelled distance.

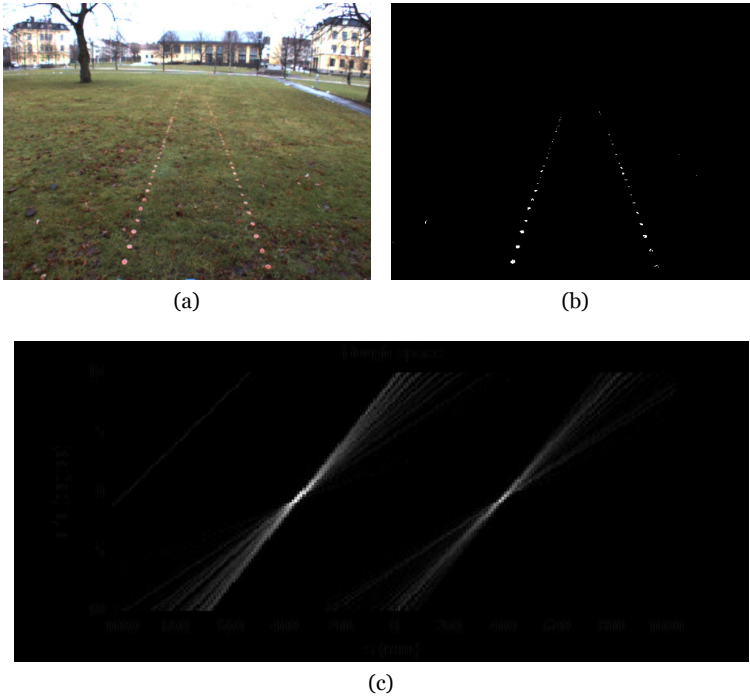


Figure 4.6: Example image from the row detection system used in Paper III. a) shows the captured color image, b) shows the binary image after color filtering and segmentation, c) shows the Hough space plotted in α -s-plane. Each maximum (white fields) represents one row.

Test	Visual odometry (m)	Ground truth (m)	Error
1	11.132	11.280	1.3%
2	10.913	11.150	2.1%
3	10.631	10.550	0.7%

Table 4.2: Result from Paper III where the robot drove 10 meters autonomously along the row. The reported distance of the visual odometry system is compared to the position measured using a ruler.

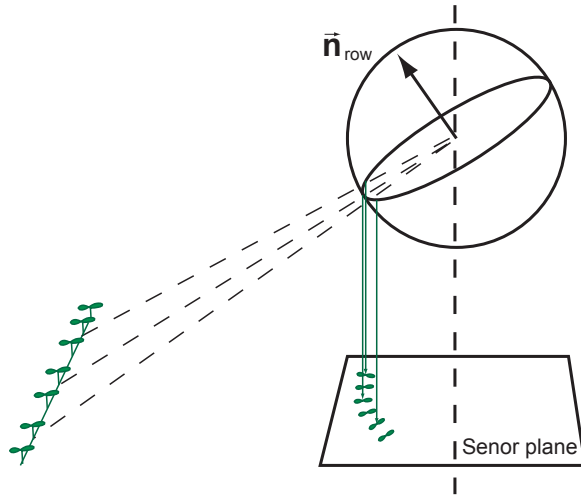


Figure 4.7: Camera model used in Paper IV: Projection through the lens on to the sensor plane by a unit sphere model of the omnidirectional camera

4.4 PAPER IV: ROW-DETECTION ON AN AGRICULTURAL FIELD USING OMNIDIRECTIONAL CAMERA

Stefan Ericson and Björn Åstrand (2010). “Row-detection on an agricultural field using omnidirectional camera”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010. IROS 2010*, pp. 4982–4987

This paper presented at the IEEE/RJS International Conference on Intelligent Robots and Systems (IROS), 2010, Taipei, Taiwan, describes a method of detecting parallel rows on an agricultural field using an omnidirectional camera. The Hough transform has shown good results in estimating rows in agricultural fields, even with high weed pressure. A forward-facing camera is the most common setup for a row-following system. A drawback of using this type of camera is that the row is lost at the end of the row, and there is nothing to track. This paper shows how to detect crop lines using an omnidirectional camera instead. The method works both on cameras with a fisheye lens and cameras with a catadioptric lens. Figure 4.7 shows the camera model used.

A combination of an edge-based method (Bazin et al., 2009) and a Hough transform method is used to both detect individual rows and to find the vanishing point of several parallel rows. The method was evaluated on synthetic images generated by a simulator that used calibration data from real lenses. Scenes with several rows were produced, with each plant positioned randomly with a specified variance. Experiments were performed on these synthetic images and on real field images. The results show that the performance of the two methods combined is better than each method by itself.

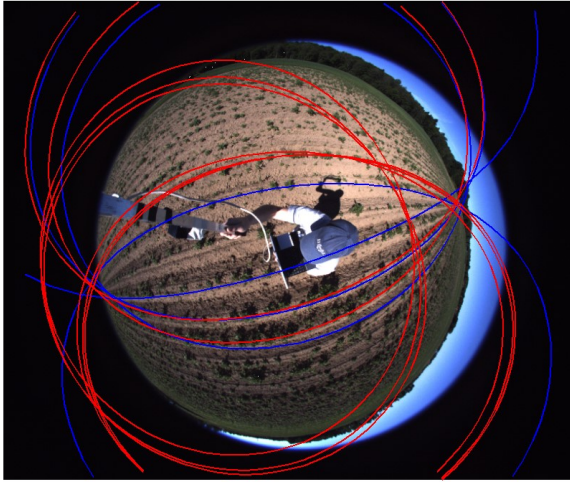


Figure 4.8: Result from Paper IV showing detected rows on a potato field captured with a fisheye camera. Blue lines show rows detected as lines, and red lines show rows detected using a Hough transform.

4.5 PAPER V: EGO-MOTION ESTIMATION BY AN AGRICULTURAL FIELD ROBOT USING VISUAL ODOMETRY

Stefan Ericson and Björn Åstrand (2016). “Ego-motion estimation by an agricultural field robot using visual odometry”. Submitted to Biosystems Engineering

This paper, submitted to the Biosystems Engineering journal, analyzes two visual odometry systems for use in an agricultural field environment. The effects of various design parameters and camera setups were evaluated in a simulated environment. The simulator was used to create data sets so that different design options could be evaluated. Four real field experiments were conducted using a mobile robot operating in an agricultural field. The robot was controlled to travel in a regular back-and-forth pattern with headland turns as shown in Figure 4.9. The experimental runs were 1.8–3.1 km long and consisted of 32,000–63,000 frames.

The results indicate that high resolution images captured from forward-facing stereo cameras mounted with a large baseline give the best results. The algorithm must be able to reduce error accumulation by adapting the frame rate to minimize error. The results also illustrate the difficulties of estimating roll and pitch using a downward-facing camera. Table 4.3 shows the results for full 6-DoF position estimation using every 5th frame in the data set. The best results were obtained on the Field 2 sequence, which is a 1.8 km run using 6680 frames captured from the forward-facing cameras. The translation error (x, y, z) is 3.76% and the rotational error (i.e., roll, pitch, and yaw) is $0.0482 \text{ deg m}^{-1}$.



Figure 4.9: The field experiment used in Paper V and Paper VI. The plot from Google Earth shows the robot path on the field for three test runs. Map data: Google, Lantmäteriet, Metria. The left-hand track in white represents the Field 1 experiment and the right-hand track in white the Field 2 experiment. The red track is the path of the Field 4 experiment.

Table 4.3: Results from Paper V. Visual odometry on an oat and fodder field with a sub-sample of the data set.

Test setup	Distance (m)	Translation error	Rotation error (deg m ⁻¹)	Matches (average)	Inliers (average)
Field 1	2093	17.1%	0.0404	70	79%
Field 2	1838	3.76%	0.0482	184	90%
Field 3	2366	3.91%	0.0365	174	83%
Field 4	3065	10.8%	0.0898	152	66%

4.6 PAPER VI: LOCALIZATION OF AGRICULTURAL FIELD ROBOTS USING NATURAL LANDMARK AND VISUAL SLAM

Stefan Ericson and Björn Åstrand (2017). “Localization of agricultural field robots using natural landmark and vision SLAM”. Submitted to Sensors

In this paper, submitted to the MDPI journal Sensors, we evaluate two visual SLAM algorithms, the feature-based ORB-SLAM and the dense LSD-SLAM, on the same data set used in Paper V. The major difference between the visual odometry algorithms evaluated in Paper V and the SLAM algorithms evaluated in this paper is that the SLAM approaches include both local optimization and the ability to detect and close loops. Both methods use pose-graphs where the nodes represent the position of keyframes and the edges the relative pose between them. Both monocular and stereo versions of the algorithms were evaluated and the LSD-SLAM was also applied to omnidirectional images. The results show that the stereo algorithms performed much better than the monocular variant of the algorithms. ORB-SLAM performed better than LSD-SLAM in three of the four sequences. Figure 4.10 shows a plot of the best result obtained by applying stereo ORB-SLAM on the Field 3 data set. The error of this test is 2.63% translation error and 0.0321 deg m⁻¹ rotational error.

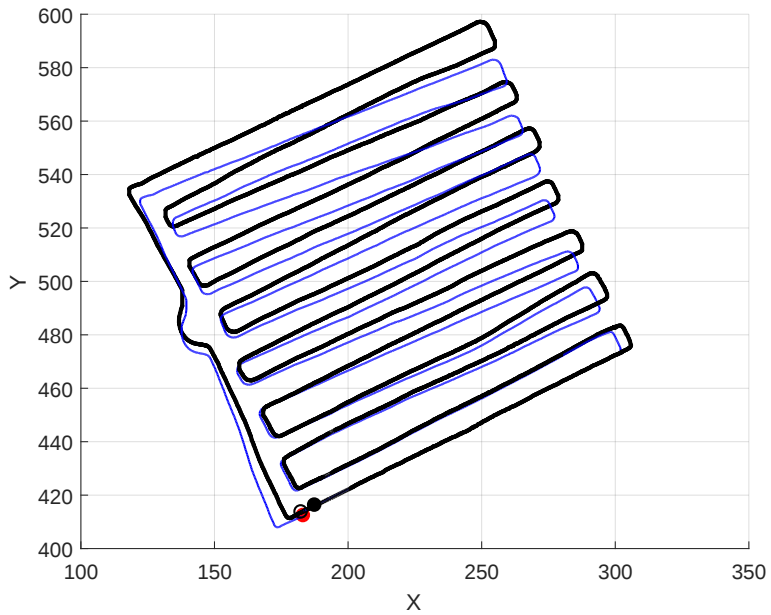


Figure 4.10: Resulting plot from using the ORB-SLAM stereo algorithm with forward-facing cameras on the Field 3 sequence; black line = ground truth, blue line = SLAM result, open circle = start position, solid circle = end position. This is the best result for all tests. There are, however, no loop closures and error accumulation still remains.

CHAPTER 5

DISCUSSION

This chapter discusses how the included papers fulfil the aims of the thesis and provide answers to the research questions. It also suggests further work within the field of visual localization in agricultural fields.

The main research question is “How and in what way can cameras be used for navigation of an agricultural field robot?” The work presented in this thesis shows how and with what accuracy cameras can be used with visual odometry algorithms to estimate relative position. Odometry can be used in different ways, either to estimate the position of a tool relative detected weed, or to estimate the position on the field given a known starting point.

This work started by evaluating visual odometry to see whether it could be used to measure short distances as a tool’s position relative to a camera. A downward-facing monocular camera was used and the algorithm estimated the position in 2-DoF, i.e., only translation in x and y . The results showed that it was possible to estimate translation, but the experiment with the camera moving back-and-forth showed less error than when it only moved forward. This indicates that the technique of visual odometry works, but also that it is sensitive to uneven ground due to the planar ground assumption. A new method was developed to both cope with uneven ground and increase robustness by introducing stereo cameras and a validation gate for matching. The NCC method was selected since it showed the best results in terms of execution time, operation range, and accuracy. Laboratory experiments showed that this method works on various types of ground such as soil, grass, and sugar beet, even though the height varied by up to 30% of the camera clearance. The error is in the order of few percent of travelled distance. For a weeding tool located one meter behind a weed detection camera, the error would be a few centimeters. This error would be fairly constant since the measurement is made relative to a reference point that moves with the camera.

Visual odometry was also evaluated for estimating 6-DoF pose on the field relative to a known start position. The experiments showed that it is very hard to estimate 6-DoF from downward-facing cameras. This is due to the high uncertainty of the depth measure. The pitch estimation showed the largest error with a clear bias since it was aligned in the travelling direction. This makes the robot lose track of the heading, leading to such large errors that this method is infeasible.

There is one experiment that failed and was not included in any paper. That ex-

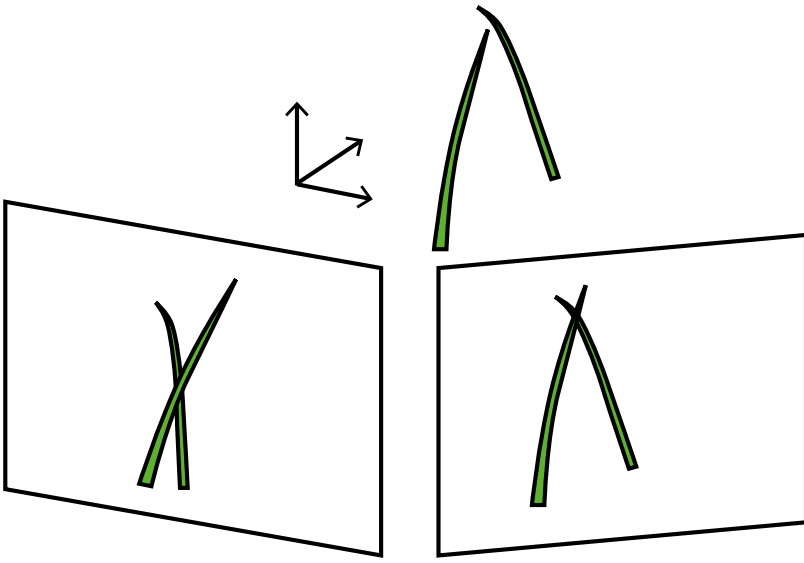


Figure 5.1: Perspective of individual blades of grass, where the trackable intersection in image plane does not represent camera motion

periment was on the fodder field with downward-facing cameras much closer to the ground. It proved to be impossible to track any points, with either dense or feature-based methods, due to the perspective change of individual blades of grass. Imagine two blades of grass at different depths, whose two-dimensional projection on the image plane makes a cross visually (Figure 5.1). By changing perspective this cross would move in the image plane, and the visual odometry algorithm tracks the cross. The problem is that the cross does not represent the motion of the camera since the new cross is a projection of other parts of the blades. Thus the position of the downward-facing camera in the experiments which are reported is about one meter above the ground. This camera placement makes it hard to use any cover to shield the area from sunlight; as a result, shadows from the robot are visible in some images. This creates an area with very low contrast where the tested feature-based approaches fail to find any features.

The results show that 6-DoF visual odometry works much better with forward-facing cameras. This is due to the presence of points far away that can be tracked over more frames and provide a better estimation of the heading. There is still drift, but it is not as severe as in the downward-facing case. The visual odometry algorithm that provided the best performance was *Libviso*, even though it lacks local optimization.

The localization result depends on many design options in the visual odometry system. First, it is very important that the motion of the robot does not violate any of the visual odometry limits. For instance, the overlap between images has to be large enough that correspondence can be found. The overlap depends on the velocity of the vehicle, frame rate, field of view, and placement of the camera. A downward-facing camera mounted close to the ground requires a high frame rate and a computationally light algorithm so that real-time performance can be obtained. The best accuracy for the 6-DoF visual odometry was obtained when the captured images were high resolution, captured from forward-facing stereo cameras, mounted with

a large stereo baseline. The major issue with visual odometry is error accumulation, which has to be dealt with if odometry is to be used for global localization on the field. The algorithm must also detect and handle large errors in the motion estimates to prevent motion outliers, since these have a large effect on the final result.

The ability to use visual odometry in navigation is improved by combining it with a row-following system. No large scale field experiment was performed, only a short test on grass with artificial rows consisting of red markers. It may seem that the red markers simplify the row detection problem, but the same steps are still required in the algorithm, i.e., color separation and segmentation followed by a Hough transform. One prerequisite of the row detection system is that the image can be segmented to separate the row from the surroundings. However, this system assumes planar ground and a camera with constant angle to the plane, which makes it sensitive to uneven terrain. The extended work demonstrates the use of cameras with an omnidirectional view to identify rows consisting of both solid lines and individual plants. When multiple lines are detected, these can be evaluated to find the vanishing point in the field, which gives a relative measurement of the robot's heading on the field. The heading is provided in 2-DoF, which overcomes the problem with a constant camera angle. This work also used simulations to evaluate the algorithm, in combination with real world experiments to verify functionality. This work also demonstrates the systematic method followed in this thesis by iteratively improving the artefact based on previous knowledge.

The limitations of visual odometry become clear when localization on a field relative to a known start position is required. The accumulated error has to be dealt with, and one way is to apply a SLAM algorithm that has the ability to both apply local optimization and detect loops and close them for improved performance. The result shows that there is a significant improvement of the result between the visual odometry algorithms and the SLAM. The main reason is that the SLAM algorithm only uses the number of frames required for a suitable amount of overlap. Thus the drift when stationary is handled correctly, and every other frame is used when travelling on a straight path. The result is more comparable with the results obtained using only every 5th frame in visual odometry. Still there are large errors, and the global consistency requirement is not always met. The reason is, firstly, that no loops were detected and closed and the full potential of SLAM was not demonstrated. Secondly, the optimization algorithm can make mistakes. This can lead to larger errors than in the case of visual odometry. An example is the LSD-SLAM algorithm which completely missed a headland turn in the experiment on Field 3.

The findings of this thesis are of practical use to developers who intend to use vision for localization in agricultural fields, particularly when it comes to developing new robots to carry out specific agricultural tasks.

5.1 FUTURE WORK

Further work need to be done to improve the SLAM algorithm. The ORB-SLAM method provided the best result so far, but it is not able to close loops when following a back-and-forth driving path with headland turns. The resulting drift in orientation cannot be compensated for, which leads to a large error. This problem could potentially be solved by separating the heading estimation into parts and using separate keyframes containing only points far away from the robot, i.e., on the horizon. That should enable loop closure at each headland turn, which might reduce the orientation error.

It would also be interesting to apply the system on other robots or tractors to evalu-

ate its performance. There have to be more field experiments to survey the robot's autonomous performance in different weather conditions and seasons.

CHAPTER 6

CONCLUSIONS

This thesis demonstrates in what way cameras can be used for localization by an agricultural field robot. A 2-DoF visual odometry system can be used with a downward-facing camera to estimate the relative positions of a crop identification camera and a tool.

A visual odometry system is based on template matching using normalized cross correlation. A novel validation gate was proposed to reduce the number of false motion estimates. This system provided an accuracy of 0.6% of travelled distance in laboratory environments over a short distance (0.7 m). This visual odometry system was implemented for real-time performance and combined with a row detection system based on a Hough transform that is applied to images from a forward-facing camera. Both systems were implemented on a mobile robot, and an autonomous drive along a 10 meter track demonstrated a position error of 2% of travelled distance.

The row detection system assumes that the camera has a constant angle to the field. An improved row detection system was developed where a Hough transform is combined with a line detection system to be able to detect crop rows consisting both of solid lines and individual plants. This detection system was applied to images from an omnidirectional camera, and were able to find the vanishing point with an improved performance compared to using either algorithm on its own.

Visual odometry can also be used for localization in 6-DoF on an agricultural field. Forward-facing stereo cameras proved more accurate than downward-facing cameras. High resolution images from stereo cameras with a wide baseline increase the accuracy. The main error source is drift, which occurs due to accumulating errors from each frame analyzed. A reduction in frame rate improves the result significantly, as long as there is still sufficient overlap for matching between the frames. Reducing the frame rate improved the visual odometry result for a 1.8 km field experiment from 8.69% to 3.76% translation error of travelled distance.

The use of the SLAM algorithm overcomes the problem of frame rate selection since both local and global optimization is performed. It also has the ability to close loops, something that never occurred on the field tests when driving in a traditional back-and-forth pattern. However, that led to drift, especially in heading angle, which made this approach show similar performance to visual odometry using an adapted frame rate. Yet applying the SLAM algorithm shows the possibility of using a cam-

era for localization on an agricultural field. A full implementation of SLAM would improve the robustness and the reliability of the existing system, which could potentially enable the development of commercial autonomous agricultural field robots.

REFERENCES

- Åstrand, Björn (2005). "Vision based perception for mechatronic weed control". PhD thesis. Chalmers university of technology.
- Åstrand, Björn and Albert-Jan Baerveldt (2005). "A vision based row-following system for agricultural field machinery". In: *Mechatronics* 15.2, pp. 251–269. ISSN: 0957-4158.
- Atherton, Kelsey D. (2016). *Autonomous Tractor Concept Takes The Farmers Out Of Farming*. Online. Popular Science. URL: <http://www.popsoci.com/autonomous-tractor-concept-takes-farmers-out-farming>.
- Bazin, J.C. et al. (2009). "Dynamic programming and skyline extraction in catadioptric infrared images". In: *Proceedings of the 2009 IEEE international conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., pp. 532–539.
- Bechar, Avital and Clement Vigneault (2016). "Agricultural robots for field operations: Concepts and components". In: *Biosystems Engineering* 149, pp. 94–111. ISSN: 1537-5110. DOI: <http://dx.doi.org/10.1016/j.biosystemseng.2016.06.014>.
- Bell, Thomas (2000). "Automatic tractor guidance using carrier-phase differential GPS". In: *Computers and Electronics in Agriculture* 25.1-2, pp. 53–66. ISSN: 0168-1699. DOI: [http://dx.doi.org/10.1016/S0168-1699\(99\)00055-1](http://dx.doi.org/10.1016/S0168-1699(99)00055-1).
- Blackmore, BS et al. (2008). "A specification for an autonomous crop production mechanization system". In: *International Symposium on Application of Precision Agriculture for Fruits and Vegetables* 824, pp. 201–216.
- Bouguet, J. Y. (2008). *Camera calibration toolbox for Matlab*. URL: http://www.vision.caltech.edu/bouguetj/calib%5C_doc/.
- Britannica, Encyclopædia (2016). *Encyclopædia Britannica*. URL: <http://academic.eb.com/levels/collegiate/article/42505>.
- Camera calibration With OpenCV* (2016). Accessed: 2017-01-05. URL: http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html.
- Cheein, F. Auat et al. (2011). "Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection". In: *Computers and Electronics in Agriculture* 78.2, pp. 195–207. ISSN: 0168-1699. DOI: <http://dx.doi.org/10.1016/j.compag.2011.07.007>.
- Cvišić, Igor and Ivan Petrović (2015). "Stereo odometry based on careful feature selection and tracking". In: *European Conference on Mobile Robots 2015*. DOI: 10.1109/ECMR.2015.7324219.
- Dewey, J., J.A. Boydston, and J. Gouinlock (2008). *The Later Works of John Dewey, Volume 2, 1925 - 1953: 1925-1927, Essays, Reviews, Miscellany, and the Public and Its Problems*. Collected Works of John Dewey Series v. 2. Southern Illinois University Press. ISBN: 9780809328123.
- Eden, Amnon H. (2007). "Three Paradigms of Computer Science". In: *Minds and Machines* 17.2, pp. 135–167. ISSN: 1572-8641. DOI: 10.1007/s11023-007-9060-8.
- Engel, J., J. Stückler, and D. Cremers (2015). "Large-scale direct SLAM with stereo cameras". In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 1935–1942. DOI: 10.1109/IROS.2015.7353631.
- Engel, Jakob and Daniel Cremers (2014). "LSD-SLAM: Large-scale direct monocular SLAM". In: *In ECCV*.

- Ericson, Stefan and Björn Åstrand (2007). "Algorithms for Visual Odometry in Outdoor Field Environment". In: *Proceedings of 13th IASTED International Conference Robotics and Applications*. Würzburg, Germany, pp. 287–292.
- Ericson, Stefan and Björn Åstrand (2008). "Stereo Visual Odometry for Mobile Robots on Uneven Terrain". In: *IAENG Special Edition of the World Congress on Engineering and Computer Science, Advances in Electrical and Electronics Engineering*, pp. 150–157.
- Ericson, Stefan and Björn Åstrand (2009). "A vision-guided mobile robot for precision agriculture". In: *Proceedings of 7th European Conference on Precision Agriculture*. Wageningen, Netherland, pp. 623–630.
- Ericson, Stefan and Björn Åstrand (2010). "Row-detection on an agricultural field using omnidirectional camera". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010. IROS 2010*, pp. 4982–4987.
- Ericson, Stefan and Björn Åstrand (2016). "Ego-motion estimation by an agricultural field robot using visual odometry". Submitted to Biosystems Engineering.
- Ericson, Stefan and Björn Åstrand (2017). "Localization of agricultural field robots using natural landmark and vision SLAM". Submitted to Sensors.
- Fiorini, Paolo and Erwin Prassler (2000). "Cleaning and Household Robots: A Technology Survey". In: *Autonomous Robots* 9.3, pp. 227–235. ISSN: 1573-7527. DOI: 10.1023/A:1008954632763.
- Fischler, Martin A. and Robert C. Bolles (1981). "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Commun. ACM* 24 (6), pp. 381–395. ISSN: 0001-0782. DOI: <http://doi.acm.org/10.1145/358669.358692>.
- Fraundorfer, F. and D. Scaramuzza (2012). "Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications". In: *Robotics Automation Magazine, IEEE* 19.2, pp. 78–90. ISSN: 1070-9932. DOI: 10.1109/MRA.2012.2182810.
- Fyfe, Aileen (2005). "J. L. Heilbron (ed.), The Oxford Companion to the History of Modern Science. Oxford and New York: Oxford University Press, 2003. Pp. xxviii+941. ISBN 0-19-511229-6." In: *The British Journal for the History of Science* 38.3, pp. 351–353. DOI: 10.1017/S0007087405227279.
- Gan-Mor, Samuel, Rex L. Clark, and Bruce L. Upchurch (2007). "Implement lateral position accuracy under RTK-GPS tractor guidance". In: *Computers and Electronics in Agriculture* 59.1-2, pp. 31–38. ISSN: 0168-1699.
- Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: 10.1109/CVPR.2012.6248074.
- Geiger, Andreas, Julius Ziegler, and Christoph Stiller (2011). "StereoScan: Dense 3d Reconstruction in Real-time". In: *IEEE Intelligent Vehicles Symposium*. Baden-Baden, Germany. DOI: 10.1109/IVS.2011.5940405.
- Grimstad, L. et al. (2015). "On the design of a low-cost, light-weight, and highly versatile agricultural robot". In: *2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pp. 1–6. DOI: 10.1109/ARSO.2015.7428210.
- Hansen, Peter et al. (2011). "Stereo visual odometry for pipe mapping". In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 4020–4025. DOI: 10.1109/IR05.2011.6094911.

- Hansen, P. et al. (2011). "Monocular visual odometry for robot localization in LNG pipes". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3111–3116. DOI: 10.1109/ICRA.2011.5979681.
- Harris, C. and M. Stephens (1988). "A Combined Corner and Edge Detector". In: *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151.
- Hartley, R. I. and A. Zisserman (2004). *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518.
- Herring, D. D. (2001). "Precision Farming". In: *NASA's Earth Observatory*. URL: <http://earthobservatory.nasa.gov/Features/PrecisionFarming/>.
- Hevner, Alan R. et al. (2004). "Design Science in Information Systems Research". In: *MIS Q.* 28.1, pp. 75–105. ISSN: 0276-7783.
- Hicks, Rob Warren and Ernest L. Hall (2000). "A Survey of Robot Lawn Mowers". In: *SPIE - The International Society for Optical Engineering*. DOI: 10.1117/12.403770.
- Hiremath, Santosh et al. (2014). "Image-based particle filtering for navigation in a semi-structured agricultural environment". In: *Biosystems Engineering* 121, pp. 85–95. ISSN: 1537-5110. DOI: <http://dx.doi.org/10.1016/j.biosystemseng.2014.02.010>.
- Howard, A. (2008). "Real-time stereo visual odometry for autonomous ground vehicles". In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 3946–3952. DOI: 10.1109/IROS.2008.4651147.
- Jähne, Bernd (2002). *Digital Image Processing*. 5th. Springer-Verlag Berlin Heidelberg New York. ISBN: 3540677542.
- Jiang, Dawei et al. (2014). "Development of a 3D ego-motion estimation system for an autonomous agricultural vehicle". In: *Biosystems Engineering* 121, pp. 150–159. ISSN: 1537-5110. DOI: <http://dx.doi.org/10.1016/j.biosystemseng.2014.02.016>.
- John Deere Autotrak (2016). URL: https://www.deere.se/sv_SE/products/equipment/agricultural_management_solutions/guidance_and_machine_control/autotrak_universal_200/autotrak_universal_200.page.
- Jørgensen, Rasmus et al. (2007). "HortiBot: A System Design of a Robotic Tool Carrier for High-tech PlantNursing". In: *Agricultural Engineering International IX*. ISSN: 1682-1130.
- Kameras als Augen des Jät-Roboters (2015). Markt & Technik.
- Kitt, Bernd, Andreas Geiger, and Henning Lategahn (2010). "Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme". In: *IEEE Intelligent Vehicles Symposium*. San Diego, USA. DOI: 10.1109/IVS.2010.5548123.
- Klein, Georg and David Murray (2007). "Parallel Tracking and Mapping for Small AR Workspaces". In: *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. ISMAR '07. Washington, DC, USA: IEEE Computer Society, pp. 1–10. ISBN: 978-1-4244-1749-0. DOI: 10.1109/ISMAR.2007.4538852.
- Klemke, E.D. et al. (1998). *Introductory Readings in the Philosophy of Science*. Prometheus Books. ISBN: 9781573922401. URL: <https://books.google.se/books?id=kzgfAQAAIAAJ>.
- Krüger, G., R. Springer, and W. Lechner (1994). "Global Navigation Satellite Systems (GNSS)". In: *Computers and Electronics in Agriculture* 11.1, pp. 3–21. ISSN: 0168-1699. DOI: [http://dx.doi.org/10.1016/0168-1699\(94\)90049-3](http://dx.doi.org/10.1016/0168-1699(94)90049-3).

- Länsstyrelsen (2016). *Flyghavre*. URL: <http://www.lansstyrelsen.se/Vastra-Gotaland/Sv/lantbruk-och-landsbygd/Vaxtodling-vaxtskydd/flyghavre/Pages/flyghavre.aspx> (visited on 12/29/2016).
- Lantmäteriet (n.d.). <https://swepos.lantmateriet.se/>. Accessed: 2016-12-29. URL: <https://swepos.lantmateriet.se/>.
- Lechner, Wolfgang and Stefan Baumann (2000). "Global navigation satellite systems". In: *Computers and Electronics in Agriculture* 25.1-2, pp. 67–85. ISSN: 0168-1699.
- Lepej, Peter and Jurij Rakun (2016). "Simultaneous localisation and mapping in a complex field environment". In: *Biosystems Engineering* 150, pp. 160–169. ISSN: 1537-5110. DOI: <http://dx.doi.org/10.1016/j.biosystemseng.2016.08.004>.
- Link, A., M. Panitzki, and S. Reusch (2002). "Hydro N-sensor: tractor-mounted remote sensing for variable nitrogen fertilization". In: *P.C Robert, R.H. Rust, & W.E. Larson (Eds.), Proceedings of the 6th International Conference on Precision Agriculture*. ASA/CSSA/SSSA. Madison, WI, USA, pp. 1012–1018.
- Lowe, David G. (2004). "Distinctive Image Features from Scale-Invariant Key-points". In: *International Journal of Computer Vision* 60 (2), pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94.
- Lucas, Bruce D. and Takeo Kanade (1981). "An iterative image registration technique with an application to stereo vision". In: *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, pp. 674–679.
- Maimone, Mark, Yang Cheng, and Larry Matthies (2007). "Two years of Visual Odometry on the Mars Exploration Rovers". In: *Journal of Field Robotics* 24.3, pp. 169–186. ISSN: 1556-4967. DOI: 10.1002/rob.20184.
- Matthies, L. and S. Shafer (1987). "Error modeling in stereo navigation". In: *Robotics and Automation, IEEE Journal of* 3.3, pp. 239–248. ISSN: 0882-4967. DOI: 10.1109/JRA.1987.1087097.
- Mei, C. and P. Rives (2007). "Single View Point Omnidirectional Camera Calibration from Planar Grids". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3945–3950. DOI: 10.1109/ROBOT.2007.364084.
- Moberg, Linn (2013). *En gräsligt bra sommar*. Mariestadstidningen. 28.
- Montemerlo, Michael et al. (2002). "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem". In: *Eighteenth National Conference on Artificial Intelligence*. Edmonton, Alberta, Canada: American Association for Artificial Intelligence, pp. 593–598. ISBN: 0-262-51129-0.
- Montemerlo, M. et al. (2003). "FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges". In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*. IJCAI. Acapulco, Mexico.
- Mur-Artal, R., J. M. M. Montiel, and J. D. Tardós (2015). "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5, pp. 1147–1163. ISSN: 1552-3098. DOI: 10.1109/TR0.2015.2463671.
- NAVSTAR GPS User Equipment Introduction (1996).
- Nistér, David (2005). "Preemptive RANSAC for live structure and motion estimation". In: *Mach. Vision Appl.* 16 (5), pp. 321–329. ISSN: 0932-8092. DOI: 10.1007/s00138-005-0006-y.

- Nistér, David, Oleg Naroditsky, and James Bergen (2004). "Visual Odometry". In: *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 1, pp. 652–659. ISSN: 1063-6919. DOI: 10.1109/CVPR.2004.265.
- Nourani-Vatani, Navid and Paulo Vinicius Koerich Borges (2011). "Correlation-based visual odometry for ground vehicles". In: *Journal of Field Robotics* 28.5, pp. 742–768. ISSN: 1556-4967. DOI: 10.1002/rob.20407.
- Oates, Briony J (2006). *Researching Information Systems and Computing*. Sage Publications Ltd. ISBN: 1412902231.
- Olson, Edwin, John Leonard, and Seth Teller (2006). "Fast Iterative Optimization of Pose Graphs with Poor Initial Estimates". In: pp. 2262–2269.
- Pierce, Francis J. and Peter Nowak (1999). "Aspects of Precision Agriculture". In: ed. by Donald L. Sparks. Vol. 67. *Advances in Agronomy*. Academic Press, pp. 1–85. DOI: DOI:10.1016/S0065-2113(08)60513-1.
- Pilarski, Thomas et al. (2002). "The Demeter System for Automated Harvesting". In: *Autonomous Robots* 13.1, pp. 9–20.
- Popper, K.R. (2002). *The Logic of Scientific Discovery*. Classics Series. Routledge. ISBN: 9780415278447. URL: <https://books.google.com.au/books?id=Yq6xeupNStMC>.
- Robson, C. (2002). *Real World Research: A Resource for Social Scientists and Practitioner-Researchers*. Regional Surveys of the World Series. Wiley. ISBN: 9780631213055.
- Rublee, E. et al. (2011). "ORB: An efficient alternative to SIFT or SURF". In: *2011 International Conference on Computer Vision*, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.
- Ruckelshausen, A et al. (2009). "BoniRob—an autonomous field robot platform for individual plant phenotyping". In: *Proceedings of 7th European Conference on Precision Agriculture*. Wageningen, Netherland, pp. 841–847.
- Scaramuzza, D. and F. Fraundorfer (2011). "Visual Odometry : Part I: The First 30 Years and Fundamentals". In: *Robotics Automation Magazine, IEEE* 18.4, pp. 80–92. ISSN: 1070-9932. DOI: 10.1109/MRA.2011.943233.
- Scaramuzza, D., A. Martinelli, and R. Siegwart (2006). "A toolbox for easily calibrating omnidirectional cameras". In: *Proc. of the IEEE International Conference on Intelligent Systems, IROS06, Beijing, China*. Citeseer.
- Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza (2011). *Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents series)*. The MIT Press. ISBN: 0262015358.
- Stachniss, Cyrill (2013). *Lecture notes Robot Mapping - WS 2013/14*. Universität Freiburg, Germany. URL: <http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/> (visited on 12/08/2016).
- Stafford, John V., ed. (2013). *Precision agriculture 13*. Wageningen Academic Publishers. DOI: 10.3920/978-90-8686-778-3.
- Triggs, Bill et al. (2000). "Bundle Adjustment - A Modern Synthesis". In: *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*. ICCV '99. London, UK, UK: Springer-Verlag, pp. 298–372. ISBN: 3-540-67973-1.
- Wang, C.M. (1988). "Location estimation and uncertainty analysis for mobile robots". In: *Proceedings of IEEE International Conference on Robotics and Automation, 1988*. Pp. 1231–1235. DOI: 10.1109/ROBOT.1988.12229.
- Yang, Jun et al. (2007). "Evaluating Bag-of-visual-words Representations in Scene Classification". In: *Proceedings of the International Workshop on Workshop*

- on *Multimedia Information Retrieval*. MIR '07. Augsburg, Bavaria, Germany: ACM, pp. 197–206. ISBN: 978-1-59593-778-0. DOI: 10.1145/1290082.1290111.
- Zaman, Munir (2007). “High resolution relative localisation using two cameras”. In: *Robot. Auton. Syst.* 55 (9), pp. 685–692. ISSN: 0921-8890. DOI: 10.1016/j.robot.2007.05.008.
- Zhang, Naiqian, Maohua Wang, and Ning Wang (2002). “Precision agriculture—a worldwide overview”. In: *Computers and Electronics in Agriculture* 36.2-3, pp. 113–132. ISSN: 0168-1699. DOI: DOI:10.1016/S0168-1699(02)00096-0.
- Zitova, B. and J. Flusser (2003). “Image registration methods: a survey”. In: *Image and vision computing* 21.11, pp. 977–1000. DOI: 10.1.1.86.8364.

INCLUDED ARTICLES

INCLUDED ARTICLES

The following articles have been published during my PhD studies or are under review.

PAPER I

Stefan Ericson and Björn Åstrand (2007). “Algorithms for Visual Odometry in Outdoor Field Environment”. In: *Proceedings of 13th IASTED International Conference Robotics and Applications*. Würzburg, Germany, pp. 287–292

PAPER II

Stefan Ericson and Björn Åstrand (2008). “Stereo Visual Odometry for Mobile Robots on Uneven Terrain”. In: *IAENG Special Edition of the World Congress on Engineering and Computer Science, Advances in Electrical and Electronics Engineering*, pp. 150–157

PAPER III

Stefan Ericson and Björn Åstrand (2009). “A vision-guided mobile robot for precision agriculture”. In: *Proceedings of 7th European Conference on Precision Agriculture*. Wageningen, Netherlands, pp. 623–630

PAPER IV

Stefan Ericson and Björn Åstrand (2010). “Row-detection on an agricultural field using omnidirectional camera”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010. IROS 2010*, pp. 4982–4987

PAPER V

Stefan Ericson and Björn Åstrand (2016). “Ego-motion estimation by an agricultural field robot using visual odometry”. Submitted to *Biosystems Engineering*

PAPER VI

Stefan Ericson and Björn Åstrand (2017). “Localization of agricultural field robots using natural landmark and vision SLAM”. Submitted to *Sensors*

PAPER I

ALGORITHMS FOR VISUAL ODOMETRY IN OUTDOOR FIELD ENVIRONMENT

Ericson, S. and B. Åstrand (2007). "Algorithms for Visual Odometry in Outdoor Field Environment".
In: Proceedings of 13th IASTED International Conference Robotics and Applications. Würzburg,
Germany, pp. 287–292.

ALGORITHMS FOR VISUAL ODOMETRY IN OUTDOOR FIELD ENVIRONMENT

Stefan Ericson
University of Skövde
Box 408, 54128 Skövde
Sweden
stefan.ericson@his.se

Björn Åstrand
Halmstad University
Box 823, 30118 Halmstad
Sweden
bjorn.astrand@ide.hh.se

ABSTRACT

In this paper different algorithms for visual odometry are evaluated for navigating an agricultural weeding robot in outdoor field environment. Today there is an encoder wheel that keeps track of the weeding tools position relative the camera, but the system suffers from wheel slippage and errors caused by the uneven terrain. To overcome these difficulties the aim is to replace the encoders with visual odometry using the plant recognition camera. Four different optical flow algorithms are tested on four different surfaces, indoor carpet, outdoor asphalt, grass and soil. The tests are performed on an experimental platform. The result shows that the errors consist mainly of dropouts caused by overriding maximum speed, and of calibration error due to uneven ground. The number of dropouts can be reduced by limiting the maximum speed and detection of missing frames. The calibration problem can be solved using stereo cameras. This gives a height measurement and the calibration will be given by camera mounting. The algorithm using normalized cross-correlation shows the best result concerning number of dropouts, accuracy and calculation time.

KEY WORDS

Agricultural applications, computer vision, visual odometry, and optical flow

1. Introduction

A robot for mechanical weed control of sugar beets was developed in [1]. The robot employs two vision systems; a forward looking camera used for row-following and a second, color-based vision system that is able to identify and locate a single crop among weed plants, see Figure 1. This vision system controls a weeding tool, mounted at the rear of the robot, which removes the weed within the row of crops. The weeding tool is located outside the field of view of the camera. To measure the traveled distance and thus keep track of the crop position, the encoder information from the robot wheels is used. Problems using encoders are that they are sensitive to slip and uneven terrain, often resulting in an overestimation of the distance traveled. To overcome these difficulties the aim

is to replace the encoders with visual odometry by utilize the camera system used for plant identification.

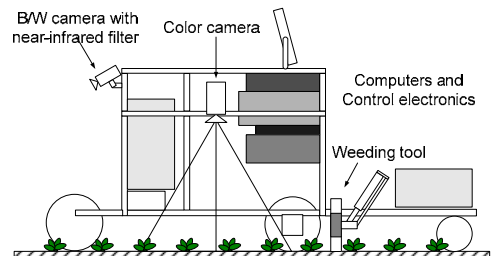


Figure 1. Mobile agricultural robot for mechanical weed control with weeding tool mounted at the rear

Using visual odometry in outdoor environment has different challenges depending on terrain. A visual odometry method that works well in one terrain may fail when applied in another. There can be a lack of coarse texture, e.g. on pavements, and loose terrain like sand, gravel or loose soil can shift under a robots wheels. Some terrains, like asphalt, may also be highly reflective. In our application the goal is to detect relative movement from consecutive images of a crop row. The problem here is that plants have different height and thus obscure the true ground plane, which may result in an error of the estimated movement.

The methods used for visual odometry in arbitrary environment can roughly be grouped into two classes; methods that compute optical flow directly on the image and methods that extract features and compute relative movement from these. Depending on application and camera arrangement, different algorithms have been proposed. For robots it is common to use either monocular camera or stereo cameras. The most common approach is to use stereo vision and extract some kind of feature from the images, and using a matching algorithm to find corresponding points in the other image. Position is obtained by tracking these points in an image sequence. A summary of previous work, with setup and performance are shown in Table 1.

Table 1. Summary of previous work

Paper	Optical flow method	Camera setup	Environment	Accuracy [x,y,θ]
Campbell et al [2]	Lucas-Kanade	Tilted mono camera heading forward	Outdoor, 3-DOF	3.3-6.1% @ 4-10m
Corke et al [3]	Lucas-Kanade	Omni-directional mono camera	Outdoor, 3-DOF	1% @ 29m
Helmick et al [4]	Forstners interest operator	Stereo	Outdoor, 3-DOF	2.5% and 5°
Nistér et al [5]	Harris corners	Stereo	Outdoor	4.86% @ 280m
Milella et al [6]	Shi-Tomasi	Stereo	Indoor, 6-DOF	2.4% @ 4m
Se et al. [7]	Scale invariant feature transform (SIFT)	3 camera L-Stereo	Indoor, 5-DOF	0.98% and 2.1° @ 4m

There are several optical flow algorithms described in the literature and nine methods are evaluated in [8]. The methods claimed to be the best for 2D translation are Lucas-Kanade method. Another evaluation of optical flow algorithms is found in [9], where a modified Lucas-Kanade method showed best result.

In [2] a tilted single camera heading forward is used. The image area is divided into a sky region and a ground region. Optical flow is computed on recent video frames using an enhanced Lucas-Kanade method. Projection of optical flow vectors into ground plane gives translation and sky region gives rotation. The result was presented for different terrain varying from 3.3% to 6.1% of distance traveled.

Another approach is to use a monocular omnidirectional camera [3]. It is used on a planetary rover together with Lucas-Kanade method [10] to track feature points in two consecutive images. Two approaches were investigated, an optical flow method and a structure to motion method, where the last method proved to be most accurate. Position estimation had an error less than 1% of traveled distance on a 29 m run.

Stereo camera approach has been used in [4] for navigating in high slip environments with a rover, using visual odometry algorithm. Feature points are obtained by using Forstner interest operator [11]. These points are matched between pairs of stereo images by using maximum likelihood estimation, and points are tracked by using a correlation-based method. The system uses an extended Kalman filter to estimate position from the visual odometry algorithm and the rover kinematics. The error of visual odometry algorithm was less than 2.5% of the distance traveled.

In [5], a stereo camera setup is used. The main idea is the same by extracting features, tracking and estimation of motion. Harris corners [12] are used as feature, tracking is performed by normalized correlation and motion estimation by preemptive RANSAC [13]. Result for the stereo algorithm varies between 1.1% and 6.3% depending on environment. This approach is also used in [6], but instead a Shi-Tomasi feature detector [14] is used. The mean error reported was less than 2.4 %.

Scale-invariant feature transform (SIFT) is a method of extracting points invariant to scale and rotation [15]. The feature points from two images are matched and the geometrical difference between the images can be calculated. This method shows very good result in different areas such as object recognition. SIFT has been used in navigation by [7].

The objective of this paper is to evaluate different optical flow algorithms for visual odometry using the monocular camera similar to what is available on the agricultural weeding robot. The performance in outdoor field environment will be evaluated with respect to robustness and real-time performance.

2. Materials and method

All experiments are performed on an experimental platform equipped with one CCD camera mounted at approximately 25 cm from the ground, see Figure 2. The camera is placed close to the centre of the robot pointing perpendicular to the ground for giving the largest optical flow for translation. A cover prevents the sun to illuminate the ground area under the camera. Instead a lamp is placed so the light condition can be controlled. This reduces error introduced by moving shadows.

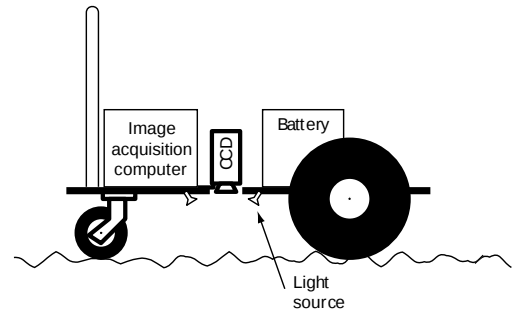


Figure 2. Experimental platform

The robot has three wheels where one is a steering wheel and the other two are intended to be driving wheels. The camera used is a Sony DFW-VL500, a FireWire CCD camera. It is configured to acquire color images with resolution 320x240 pixels at a frame rate of 15 frames per second. There is a zoom objective that is set to 552 during tests, which makes an image cover approximately

13x10cm of the ground. As reference of distance traveled and for calibration of the camera a steel scale is placed under the robot so a part of the image is showing the scale, see Figure 3. A laptop computer acquires videos from the camera. The acquired images are then analyzed offline in MATLAB©, using a Pentium D 3.0 GHz PC with 2 GB memory.

Four different methods for calculating the optical flow are tested. Normalized cross-correlation (NCC), Lukas-Kanade, Harris corners and SIFT. The first two is pixel based method, and the other two extract features and performs matching. The methods are compared and different types of errors are identified. The algorithms provide traveled distance measured in a number of pixels. To convert that number to meters, it is necessary to know the extent each pixel corresponds to. This is referred to as calibration and is measured in the first frame of a test. It is assumed that the distance between the camera and the ground is constant.

Normalized cross-correlation [16] uses two consecutive images, which are normalized to minimize influence of different illumination. The cross-correlation coefficient is calculated, which gives a similarity measurement. The point where the maximum similarity is found is used for calculating translation between the images. A matching window is selected in the center of the first image which is matched to a search area in the second image. The search area has to be larger than the matching area so a perfect match can be found for maximum expected translations in any direction. This method gives a discrete result and hence there is a discretization error. The similarity measurement gives also a confidence of the match, which can be used in a sensor fusion system. A limitation is that the method only handles translation in x , y , not scale or rotation.

The Lucas-Kanade method [10] assumes the same velocity for all pixels in a neighborhood and that intensity is conserved. The spatial intensity gradient of two images is used for setting up the gradient constraint equation, which is solved by minimizing the least square error. An iterative method is enhancing the estimate. This method gives sub-pixel accuracy, and is best suited for small changes between images (± 1 pixel). This method can solve all linear image transformation such as affine, for calculating translation and rotation. Gaussian pyramid [16] can be used to increase the range so bigger differences can be handled.

The method using Harris-corners [12], extract this feature and match points between two images by comparing neighborhoods. Distance is obtained by majority voting, which gives a discrete result.

SIFT [15] is a method that provides feature points that are invariant to scale, rotation and location. A pyramid with difference of Gaussians images is created and feature points are extracted by finding local extremas. The points

are described using an invariant descriptor and matched. This method uses interpolated values and hence gives an analog value. However, this method is computational heavy and therefore some of the code has been compiled, to increase speed.

A test is performed by moving the robot 1m along the steel scale and then back to the start position. Data is analyzed in two ways, first from start position to the one meter mark of the scale, and secondly the entire test run, where the first and the last image are compared. First test will show how accurate the system can measure translation in meters, i.e. error of traveled distance. The second test will remove all biased errors such as calibration errors, under the assumption that the bias is constant over a test run. There is also a test with stereo cameras to evaluate ground flatness.

The estimation of the robot pose $\mathbf{Z}$ is given by:

$$\mathbf{Z}(k+1) = \begin{pmatrix} x(k+1) \\ y(k+1) \\ (k+1) \end{pmatrix} = f(\mathbf{Z}(k), \mathbf{U}(k)) = \begin{pmatrix} x(k) + d(k) \cos(\theta(k) + \alpha(k)) \\ y(k) + d(k) \sin(\theta(k) + \alpha(k)) \\ \theta(k) + \alpha(k) \end{pmatrix} \quad (1)$$

There d is the distance traveled measured by the visual odometry, and θ is the robot heading and α is the change of robot heading. $\mathbf{U}$ is the measurement matrix.

The variance or covariance, i.e. the robot pose uncertainty is given by:

$$\Sigma_{k+1|k} = \nabla f_x \Sigma(k|k) \nabla f_x^T + \nabla f_U \mathbf{U}(k+1) \nabla f_U^T \quad (2)$$

There ∇f_x and ∇f_U are the Jacobian matrixes with respect to the uncertain variables of the state transition matrix, i.e. x , y , d , θ and α .

Since there will be a row following system present in the final system [1], i.e. the uncertainty of the robot heading angle θ and change of heading α will be bounded by the row following system. These experiments will only consider translation in the direction of the scale, i.e. x direction. Therefore we estimate the distance as:

$$D = d_1 + d_2 + \dots + d_N = \sum_{k=1}^N d(k) \quad (3)$$

$$E[D] = \sum_{k=1}^N E[d(k)] \quad (4)$$

$$V[\hat{D}] = N\sigma_d^2 \quad (5)$$

3. Experiments and results

Data are acquired from test runs on carpet, asphalt, grass and soil. The latter two are considered to be similar to those situations expected in field. Figure 3 shows the first

images in each measurement. The reference steel scale is placed to the left and the matching area is selected as close as possible to the center to minimize lens distortion.

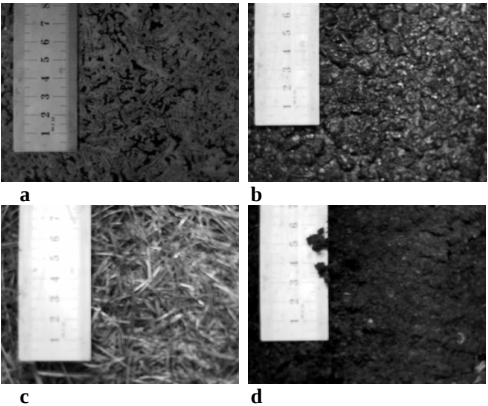


Figure 3. First image in video sequence. Steel scale for calibration and measure distance traveled (to the left)
a Indoor carpet; b Outdoor asphalt; c Outdoor grass; d Outdoor soil

Each measurement is calibrated by counting the number of pixels on a distance measured by the steel scale in the first image. The distance is between 40mm and 60mm depending on the location of the steel scale. The calibration values are between 0.36 mm/pixel to 0.40 mm/pixel for all measurements.

The reference distance is computed by comparing the position where the steel scale starts to the position marked 100 cm in an image. Examples of images are shown in Figure 4. The difference in pixels is multiplied with the calibration constant to obtain the distance traveled between the first image and the image at 100cm.

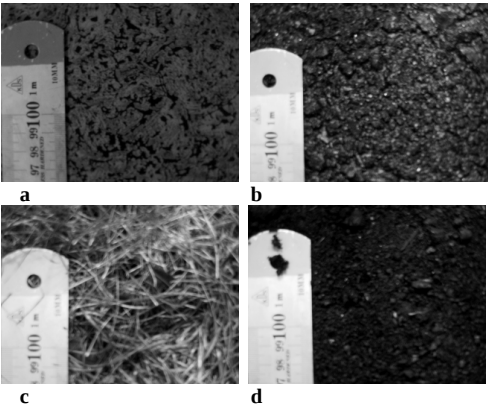


Figure 4. Images at 100 cm. a Indoor carpet; b Outdoor asphalt; c Outdoor grass; d Outdoor soil

The translation between two consecutive images depends on the speed. Maximum speed for the test runs varies between 13 and 35 cm/s corresponding to a translation of 23 to 58 pixels/frame. If a frame is dropped during image acquisition, then twice the distance will be needed. The number of images in each sequence is from 300 to 450 depending on average speed.

Figure 5 shows a typical result from a test run indoors. It shows the number of pixels moved in x-direction between two consecutive frames as a function of frame number, i.e. speed in x-direction. The Harris method shows several clear dropouts at about frame 100. NCC and SIFT shows almost the same result. A close-up, Figure 6, shows that the NCC and Harris gives a discrete signal while SIFT gives an analog.

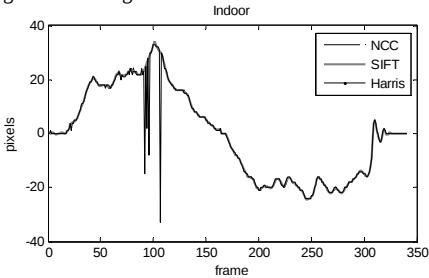


Figure 5. Typical results from indoor test run, using NCC, Harris and SIFT method

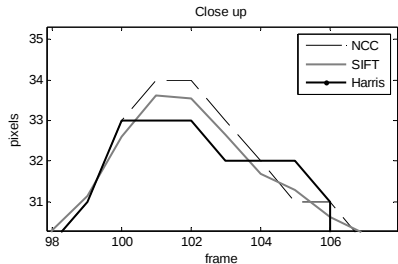


Figure 6. Close-up of indoor test run

The Lucas-Kanade method can only handle small changes, and maximum speed for current setup is 16 pixels. The method is tested on a slow run on grass. Maximum speed is 9 pixels/frame, and the result is shown in Figure 7.

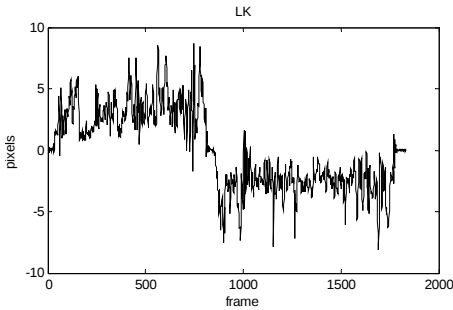


Figure 7. Result from a slow run outdoor on grass, using Lucas-Kanade method

There are 1833 frames in this run and no dropouts are detected for any algorithm. The result is shown is Table 2.

Table 2. Result from a slow test on grass, analyzed with all methods

Test Method	Error 1m	Error back & forth	Time (ms)
NCC	-1.7%	0.6%	144 ms
Lucas-Kanade	-1.4%	0.2%	302 ms
Harris corner	1.1%	3.1%	323 ms
SIFT	-1.2%	0.3%	586 ms

The errors are quite similar for NCC, Lucas-Kanade and SIFT method. Harris method stands out with poor result, probably because of dropouts that are too small to detect. Lucas-Kanade shows good result, but the limited maximum speed for a certain image resolution is a drawback.

Table 3 shows the result from four test runs on different surface. Each test is evaluated with three methods and the error is shown on percent of traveled distance. Second value shows the error for moving one meter and then back to the start position. Tests without dropouts are selected.

Table 3. Error of traveled distance for 1m test run back & forth.

Test Method	Indoor carpet	Outdoor asphalt	Outdoor grass	Outdoor soil
NCC	0.2% 0.1%	-0.6% 0.0%	-7.5% 0.9%	-3.3% 0.4%
Harris corner	0.6% 0.5%	-0.3% 0.2%	-7.5% 0.8%	-3.2% 0.5%
SIFT	0.2% 0.1%	-0.5% 0.1%	-7.6% 0.6%	-3.7% 0.2%

Both outdoor test on grass and soil showed large biased errors. It depends on that the scale in the first frame is not on the same height as the ground texture, and then a calibration error occurs.

The mean error and variance are calculated for five indoor test runs. Tests without dropouts are selected. The result is shown in Table 4.

Table 4. Mean error and variance

Test Method	Mean Error	Variance
NCC	-0.25%	0.15
Harris corner	-0.14%	0.29
SIFT	-0.49%	0.25

The number of dropouts is counted for five tests runs on each surface. Average number of frames is 350 for each run. The result is shown is Table 5.

Table 5. Number of dropouts and mean calculation time per frame for 5 different test runs

Test Method	Indoor carpet	Outdoor asphalt	Outdoor grass	Outdoor soil
NCC	0 drops 174ms	2 drops 166ms	5 drops 173ms	0 drops 181ms
Harris corner	6 drops 674ms	9 drops 473ms	6 drops 812ms	2 drops 783 ms
SIFT	0 drops 1337ms	2 drops 1588ms	7 drops 1183ms	2 drops 752ms

Analyzing the dropouts, it can be seen that the dropouts are generated in the following cases:

- Speed above maximum allowed.
- Frame missing, and speed is higher than half the maximum speed
- Texture mismatch.
- Fast accelerations. Robot motion actually acts like a dropout.

The calculation time is too long for real time application for all methods since there are only 66ms between the images at 15 frames/sec. SIFT method is very computational intense, and the time depends on the number of points detected. Even though some part of the code is compiled, the speed is not competitive to the other methods. The calculation time for the other methods can be decreased by optimizing and compiling the code.

As said before, the outdoor test on grass and soil showed a large biased error. Therefore a third test is performed to evaluate ground flatness. The same run is performed with a stereo camera setup, and the data is analyzed for finding difference in camera height. Figure 8 shows the result.

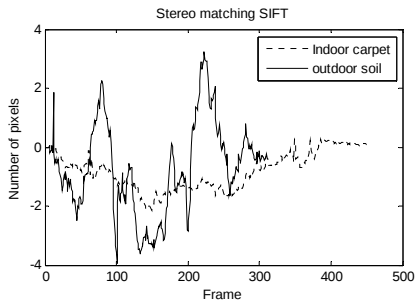


Figure 8. Flatness of ground

Indoor carpet shows only small change with a stereo camera setup, while outdoor soil case shows a much more varying surface. It can be seen that the height should be symmetrically around the center frame (150 for soil case), due to driving back and forth on the same place. But there is a little difference, which comes from the method used for moving the robot.

4. Conclusion

Optical flow by normalized cross-correlation has shown to give a good measurement of traveled distance. It performs a fast measurement with predictable calculation time, which is good for real time system. The number of dropouts is low.

The error consists mainly of dropouts and calibration errors. The discretization error can be neglected for running speeds much larger than one pixel per frame. To avoid dropouts, maximum speed had to be limited, so the vision algorithm works within range. Missing frames can easily be detected and compensated for. The calibration error can be very large for uneven surface. There is no major difference, other than calibration problems, between surfaces as long as the texture is good enough for tracking. It is noticeable for the soil case that there are fewer points to track. A solution to the calibration problem is to use stereo cameras and epipolar geometry to measure the distance to the ground.

However, visual odometry can be used for replacing the encoder wheel, since the error is small for short distance relative movements. Further work is to incorporate the row-following system and to implement it on the robot. Another goal is to extend the navigation system on the robot with other sensors such as GPS and use sensor fusion for estimating an absolute position.

References

[1] B. Åstrand & A.J. Baerveldt, Vision-based perception for an agricultural mobile robot for mechanical weed control, *Autonomous Robots* 13, 2002, 21–35.

[2] J. Campbell, R. Sukthankar, I. Nourbakhsh, & A. Pahwa, A Robust Visual Odometry and Precipice Detection System Using Consumer-grade Monocular Vision. *Proceedings of International Conference on Robotics and Automation*. Barcelona, Spain, 2005, 3421–3427.

[3] P. Corke, D. Strelow, & S. Singh, Omnidirectional visual odometry for a planetary rover. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sendai, Japan, 2004, 4007–4012.

[4] D.M. Helmick, Y. Chang, S.I. Roumeliotis, D. Clouse, & L. Matthies, Path Following using Visual Odometry for a Mars Rover in High-Slip Environments. *IEEE Aerospace Conference Vol 2*, 2004, 772–789.

[5] D. Nister, O. Naroditsky, & J. Bergen, Visual odometry. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA, 2004, 652–659.

[6] A. Milella, & R. Siegwart, Stereo-Based Ego-Motion Estimation Using Pixel Tracking and Iterative Closest Point. *IEEE International Conference on Computer Vision Systems (ICVS'06)*. New York, NY, USA, 2006.

[7] S. Se, D. Lowe, & J. Little, Vision-based Mobile robot localization and mapping using scale-invariant features, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, 2001, 2051–2058.

[8] J.L. Barron, DJ Fleet, S.S. Beauchemin, & T.A. Burkitt, Performance Of Optical Flow Techniques. *IEEE Conference on Computer Vision and Pattern Recognition*, 1992, 236–242.

[9] B. Galvin, B. McCane, K. Novins, D. Mason, & S. Mills, Recovering motion fields: An evaluation of eight optical flow algorithms. *Proceedings of the Ninth British Machine Vision Conference (BMVC '98)*, Southampton, UK, 1998, 195–204.

[10] B.D. Lucas, & T. Kanade, An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*. Vancouver, BC, Canada, 1981, 674–679.

[11] R. Deriche, & G. Giraudon, A Computational Approach for Corner and Vertex Detection. *International Journal of Computer Vision*, 10(2), 1993, 101–124.

[12] C.G. Harris, & M. Stephens, A combined corner and edge detector. *Proc. 4th Alvey Vision Conference*. Manchester, UK, 1988, 147–151.

[13] D. Nister, Preemptive RANSAC for live structure and motion estimation. *IEEE International Conference on Computer Vision*. Nice, France, 2003, 199–206.

[14] J. Shi, & C. Tomasi, Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*. Seattle, WA, USA, 1994, 593–600.

[15] D.G. Lowe, Object Recognition from Local Scale-Invariant Features, *Proceedings of the International Conference on Computer Vision*, Corfu, Greece, 1999, 1150–1157.

[16] B. Jähne, *Digital image processing* (Berlin: Springer, 2002).

PAPER II

STEREO VISUAL ODOMETRY FOR MOBILE ROBOTS ON UNEVEN TERRAIN

S. Ericson and B. Åstrand (2008a). "Stereo Visual Odometry for Mobile Robots on Uneven Terrain".

In: IAENG Special Edition of the World Congress on Engineering and Computer Science, Advances
in Electrical and Electronics Engineering, pp. 150–157.

©2008, IEEE. Reprinted, with permission, from S. Ericson & B. Åstrand

Stereo Visual Odometry for Mobile Robots on Uneven Terrain

Stefan Ericson

School of Technology and Society
University of Skövde
Skövde, Sweden
stefan.ericson@his.se

Björn Åstrand

IDE
Halmstad University
Halmstad, Sweden
bjorn.astrand@hh.se

Abstract— In this paper we present a stereo visual odometry system for mobile robots that is not sensitive to uneven terrain. Two cameras are mounted perpendicular to the ground and height and traveled distance are calculated using normalized cross correlation. A method for evaluating the system is developed, where flower boxes containing representative surfaces are placed in a metal-working lathe. The cameras are mounted on the carriage which can be positioned manually with 0.1 mm accuracy. Images are captured every 10 mm over 700 mm. The tests are performed on eight different surfaces representing real world situations. The resulting error is less than 0.6% of traveled distance on surfaces where the maximum height variation is measured to 96 mm. The variance is measured for eight test runs, total 5.6 m, to 0.040 mm. This accuracy is sufficient for crop-scale agricultural operations.

Keywords—Agricultural applications; Image processing; Mobile robot localization; Visual odometry

I. INTRODUCTION

There are many sensors available for mobile robot localization. Common ones are wheel encoders, GPS, inertial measurement unit (IMU) and machine vision [1],[2]. All these sensors have some individual drawbacks that limit the environment in where they can operate. A GPS sensor requires that the robot is outdoors and that there is a clear path to the satellites. Wheel encoders assume the wheel diameter is constant and that no wheel slippage occurs. Therefore these are used in combination to provide a global position estimate that minimizes each sensor's limitation. The position estimate can be fairly accurate if expensive sensors are used.

The global position does not always have to be that accurate, especially not for robots operating on an agricultural field on crop-scale. In that case the position relative to the crops is more important. This requires an ability to navigate locally at high precision [3]. An example of this is a robot having a camera sensor detecting and classifying crops, and an active tool performing individual plant operations out of sight of the camera, Fig. 1. The position measurement of the tool relative to the camera has to be done with high accuracy, to perform close to crop operations. Further there is an advantage to keep a tool, like a mechanical weed tool, away from the camera sensor to



Figure 1. An agricultural mobile robot for mechanical weed control on a sugar beet field [3].

prevent dust and soil from covering the lens. In these situations there is a need for a non-contact sensor that can measure short distances with high precision.

A good GPS sensor (RTK-GPS) can provide as low as 1 cm error of horizontal position [4], but only under good conditions. Error sources are the numbers of satellites visible, signal multi-path and dropouts. This position is also global, and not directly related to the crops measured in the images from the camera.

A camera sensor and an algorithm comparing consecutive frames, also known as visual odometry, can provide this non-contact measurement. There are different methods to determine the position. Work on visual odometry has been done with both mono and stereo cameras and with cameras mounted at different angle to the ground [5]-[9].

The feasibility of a visual odometer system has been shown in [5], where high slip terrain on Mars surface makes it hard to navigate using wheel encoders. Since there is no system for global positioning such as the GPS system on Mars, the visual odometry has become a critical vehicle safety system. A stereo camera setup is used and Harris corners are detected as features, which are matched and

tracked using RANSAC with least square estimator for outlier rejection.

Other work using forward looking stereo cameras for visual odometry are presented in [6]. Reported error was less than 3% without filtering and less than 1% in combination with other sensors (IMU).

In [7] different algorithms are evaluated for visual odometry using a single camera pointing perpendicular to the ground. This method assumed planar ground, which makes this solution sensitive to changes in the height of the camera above the ground. Reported error was 0.3% on planar ground and up to 7.7% on uneven grass due to calibration error.

Similar work has been done in [8] where a sensor used in optical mouse is adapted for an outdoor environment on a mobile robot. The camera and image processing is incorporated in a chip which gives a frame rate at 6 kHz. Two sensors are used for calculating heading and traveled distance, but each sensor assumes planar ground. This assumption also makes the system sensitive to tilt of the robot where one sensor is closer and the other is further away from the ground. Reported error after calibration on a straight planar surface was 0.3%.

In [9] two cameras are used, also mounted on each side of the robot. Each camera are measuring distance as a mono camera using phase correlation technique and the system are analyzed from an odometry perspective obtaining position and heading. The error reported was similar to wheel odometry.

The objective with this paper is to demonstrate the feasibility of a system for high precision stereo visual odometry in agricultural field environment.

The contribution of this paper is an odometry system for uneven terrain and an evaluation of the feasibility by experiments on textures expected to be found in real field situations.

II. MATERIAL AND METHOD

A. Method

A stereo camera setup is selected to manage the uneven terrain. The major drawback of mono camera system, including systems based on optical mice chips, is the assumption that the ground is flat. Usually the clearance between the camera and the surface is held constant by mechanical design, as in an optical mouse. This method is not suitable for an outdoor mobile robot on an agricultural field. A stereo camera setup and epipolar geometry provides a measurement of the distance between the camera and to the point evaluated. This is used to compensate for the errors caused by differences in heights. This method will also provide an easy way to calibrate the system. Only one stereo pair is analyzed so the heading of the robot is assumed to be known. Methods for obtaining direction using by omnidirectional camera is explained in [10] or by a row following system in [3].

The cameras are mounted perpendicular to the ground according to Fig. 2. This mounting will provide maximum translation between images and hence no projection of

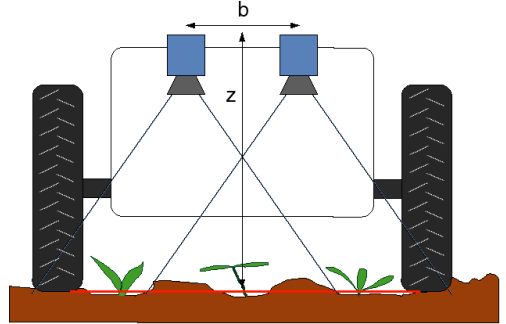


Figure 2. Camera configuration on robot. The 2D distance is measured with respect to a plane parallel to the camera lenses.

images to a ground plane is needed. Further the position is only measured in 2D.

This method is not suitable for front looking cameras, since the motion field is not perpendicular to the camera. Then some kind of perspective transformation is needed which is not included in this algorithm. The images will also capture change of heading, which is not desirable from this sensor. In the case of down-looking cameras there will be rotation around the camera axis when the heading is changed. This change will be small as long as there is good texture and the frame rate is high. If the robot drives over a rock with one of the wheels the tilt will be detected as a motion to the side. When the robot tilts back there will be an equal motion to the other side.

The use of stereo camera on a moving robot requires good synchronization. Any delay between the images will generate an error if the robot is moving. Speeds in the direction though the camera centers will be detected as a change of height and speed in the other direction will make an offset of the epipolar line. Therefore the cameras are synchronized by a hardware trigger. To reduce any errors caused by synchronization, the pair of stereo cameras should be mounted perpendicular to the main heading of the robot as shown in Fig 1, since translation in the other direction only occur when the wheel slips.

Considering a stereo camera setup [11] according to Fig. 3, the distance to a point can be calculated using (1).

$$z = \frac{bf}{d} \quad (1)$$

where z = distance from camera to point (mm)
 b = baseline (mm)
 f = focal length (pixels)
 d = disparity (pixels)

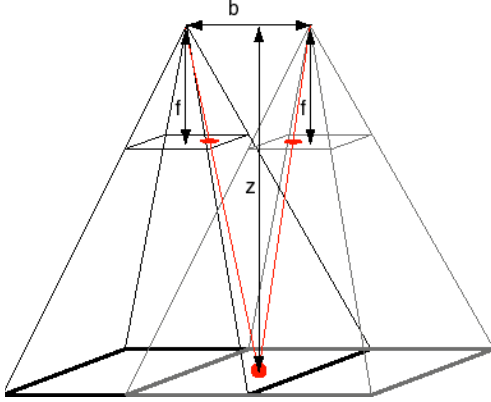


Figure 3. Camera model.

B. Matching

As shown in [7] several methods exist for analyzing the translation between consecutive images. If feature points are extracted, then some kind of matching algorithm is required to calculate translation. In [12] Harris corners are extracted and in the matching are performed with normalized cross correlation over an 11x11 pixel neighborhood. Usually normalized cross correlation is used on small regions. Two reasons are less computational cost and fewer problems caused by different projections of the images. The perpendicular mounting of the cameras and selection of a lens with low distortion reduces these problems.

Since normalized cross correlation can provide a measure of traveled distance directly without any feature extraction, this method is selected. Sub-pixel accuracy is obtained by fitting a paraboloid to the peak in the similarity image [13]. This algorithm uses a 3x3 window with the best match in the center and an elliptic paraboloid function is fitted over these 9 values. The position with sub-pixel accuracy can then be calculated from the location of the paraboloid maximum. First similarity between the stereo pair is calculated to obtain the height and then between two consecutive images of the left camera to obtain the translation. Fig. 4 shows how the disparities are calculated.

The disparity d_{zx} depends on the height to the object which is of interest. Disparity d_{zy} depends on camera alignment and camera synchronization. In a calibrated system this disparity should be zero, which means that the search area in the right image can be limited. Disparity (d_x, d_y) is the measurement of traveled distance.

Images with different heights in the same matching window are expected in a field environment. Stereo matching of a region will average the heights over that region. The same region is then used for matching distance, as shown in Fig. 4. This means that the height is the same for both matches. Therefore is the distance traveled is height independent.

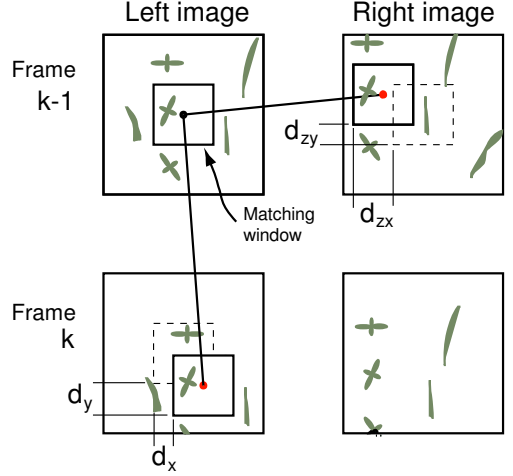


Figure 4. Method of matching images for height and distance. k is frame number and d is disparity.

From (1) the height can be calculated as:

$$z = \frac{b \cdot f}{d_{zx}} \quad (2)$$

Assuming that the height are equal for the stereo match and distance match the relation can be written as:

$$z = \frac{x \cdot f}{d_x} = \frac{b \cdot f}{d_{zx}} \quad (3)$$

which gives:

$$\mathbf{x}(k) = \mathbf{dx}(k) \cdot \frac{b}{d_{zx}(k)} \quad (4)$$

where $\mathbf{x}(k)$ = traveled distance (x, y) (mm)

$\mathbf{dx}(k)$ = disparities (d_x, d_y) (pixels)

The relation between maximum speed, camera parameters and pixel resolution are shown in (5).

$$v_{y, \max} = fps \cdot \frac{z_{\min} \cdot d_{y, \max}}{f} \quad (5)$$

where $v_{y, \max}$ = maximum speed in y direction (m/s)
 fps = framerate (frames/sec)

C. Validation Gate

The result from the normalized cross correlation is a similarity image. A perfect match is represented by a peak with maximum value one and the coordinates represents the disparities. However, this match is not always unambiguous. Region matching by normalized cross correlation suffer from the aperture problem, which occurs when there are no change in the pixel values in the direction of the motion [14]. This is caused by lack of texture on the image. In these situations the similarity image obtained from the matching will provide a bank with exactly the same values on the top.

The peaks are detected by evaluating the neighborhood to all point. A point is classified as a peak if all surrounding values are lower than the center point in, first among adjacent pixels and then among all pixels within a certain range from the center. This gives a circular area with a radius which also means that peaks can not be detected if they are close to the edge of the image.

The outcome from the peak detection gives three different cases:

- No peak is found
- One peak is found
- Several peaks are found

If no peak is found at all, the validation gate selects the point with highest value in the similarity image. This point is not detected as a peak, but it is the best guess that can be done based only on the similarity image. If only one peak is detected this peak will be selected as the best match and the position will be used for calculating the disparities.

When there are several peaks, only peaks with maximum values close to the highest peak are considered as candidates. To select the peak that most probably represents the new position, the speed is assumed to be constant. Then the position of previous match can be used for selecting the best candidate. Each peak is weighted with a Gaussian function of the distance from its position to the position of the last match (6). This will act as a filter since a match at the same position as the previous match will get a higher score.

$$w_i = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{\|\mathbf{p}_i(k) - \mathbf{p}_{\max}(k-1)\|^2}{2\sigma^2}\right) \quad (6)$$

where $\mathbf{p}_i(k)$ = position of peak (x,y) at sample k (pixels)
 i = index of peak
 w_i = weight of peak i

In the case of aperture problem the peak on the bank closest to previous match is selected.

D. Experimental Setup

A method of testing the proposed system was developed. Two USB cameras, MV BlueFOX-120aC, are mounted on



Figure 5. Test setup in lathe with cameras and flower box (grass and sugar beet).

the carriage of a metalworking lathe, as shown in Fig. 5. Then they can be moved linearly in one direction over approximately one meter with high precision. This setup provides a ground truth, something that usually is difficult to obtain in an outdoor field environment. A realistic surface is created by using flower boxes with typical crops, in this case sugar beet. These boxes can be fit into the lathe.

The carriage is positioned on the start position at one end of the flower box. Image pairs are captured while the carriage is moved to the other end of the flower box. This gives a total distance of 700 mm in y-direction for each test. The space in x-direction is so limited that no tests were possible to make in this direction.

Two different test methods are used, first for accuracy and variance calculation, which require reference of each image captured, and second for evaluating the dynamic performance. For the first method, an image pair from the stereo cameras is acquired and the carriage is moved 10mm for acquiring next pair. The cameras are not moved while the images are captured, which will eliminate errors from camera synchronization and motion blur. By capturing images at constant distance, the speed will be constant and the position for each image will be known. A drawback of this method is that the variance in the manual positioning each 10 mm will be included in the result. The accuracy of the manual positioning is estimated to be better than 0.1 mm for 95% of the measurements. This corresponds to a variance of $\sigma^2=0.0025$ mm. The mean error will be small due to the long distance.

For the evaluation of the dynamic performance, a constant frame rate is used. The carriage is moved a specified distance, in this case 700 mm, with various speeds. That means this test will include both synchronization error and motion blur, but there will not be a reference for each image, only for the entire distance. All tests are done by recording data and the analyses are performed offline using Matlab.

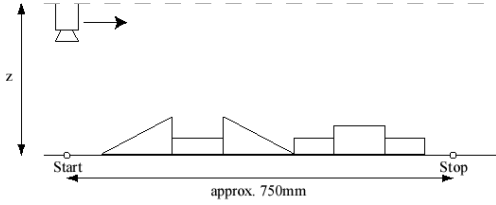


Figure 6. Height profile of reference track 2.

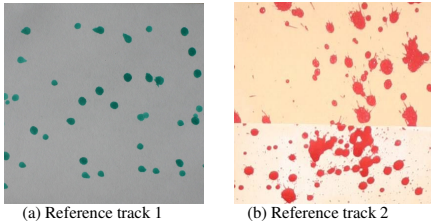


Figure 7. Texture of the two reference tracks. The image of reference track 2 is taken at one steep slope, which causes the color shift.

III. EXPERIMENTS

Three different tests runs are performed to evaluate the performance of the system. The first test is on two reference tracks and is used for calibration. The second test is on various real surfaces consisting of soil and sugar beets for variance calculation and the third test is on one selected surface to evaluate dynamic performance.

The first reference track is flat and the other consists of

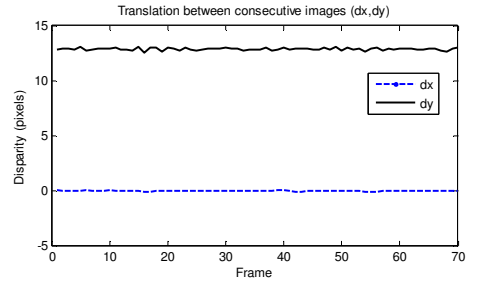


Figure 8. Translation dx , dy on reference track 1.

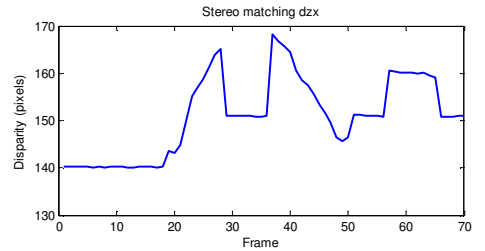


Figure 9. Disparity d_{zx} on reference track 2

linear slopes and some blocks according to Fig. 6. The height to the reference level is measured to $z = 453$ mm, and the height of the camera is 96 mm which gives a clearance of 357mm from the ground to the lens. The texture is artificial with light background and dark dots. Fig. 7 shows the surface texture of the two reference tracks.

The result from the tests on the reference track is shown

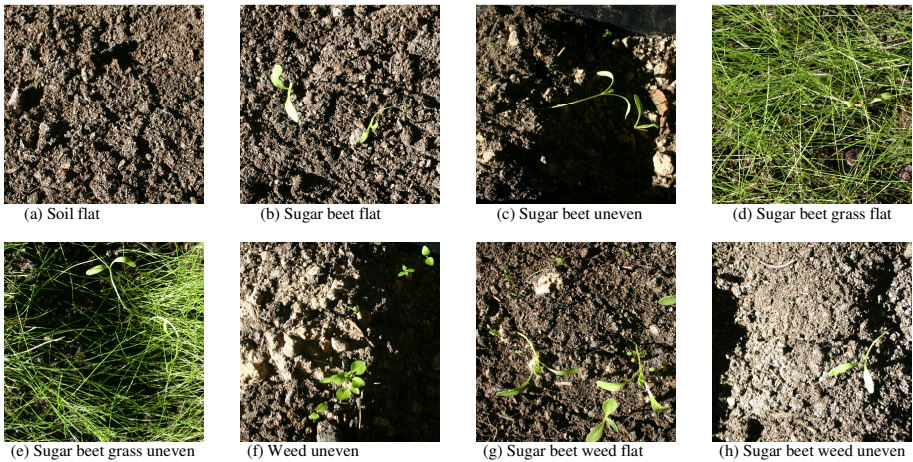


Figure 10. Texture examples of the eight flower boxes using high resolution camera.

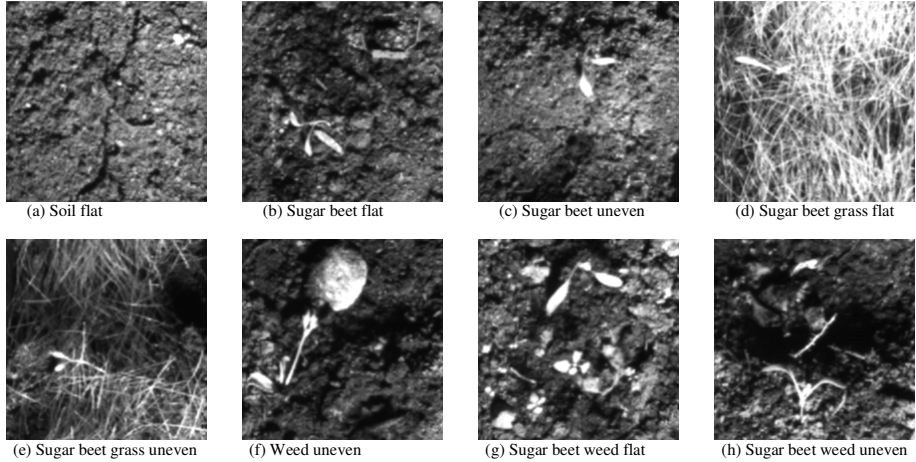


Figure 11. Texture examples of the eight flower boxes using BlueFOX camera. Image size 200x200 pixels for illustration of texture.

in Fig. 8 and Fig. 9. It can be seen that the translations measured on reference track 1 are constant with some noise, and on the reference track 2 the stereo disparity follows the height profile of the track, according to Fig. 6.

There are eight tracks prepared for tests on real surfaces. The texture is selected to cover different difficulties. Four of them are on flat surface, which means there are no intentional slopes made. These tracks are created by flattening the soil by hand. The other four are with uneven surface. These tracks are created by shaping the profile with approximately three bigger holes. The maximum height difference on a track is approximately 100 mm. Three main textures are selected; sugar beet in soil, sugar beet in soil with weed, sugar beet in grass. There are also one track on flat soil and one on uneven soil with weed. The different test surfaces are shown in Fig. 10 taken with a high resolution camera, and Fig. 11 as seen from the cameras used for image processing.

TABLE I
RESULT OF TESTS ON 700MM

Test	Measured distance y (mm)	Error (%)	Variance (mm)	Height variation (mm)
a. Soil flat	702.5	0.36	0.0084	14
b. Sugar beet flat	695.9	-0.59	0.057	24
c. Sugar beet uneven	699.9	-0.018	0.013	77
d. Sugar beet grass flat	702.1	0.31	0.024	27
e. Sugar beet grass uneven	704.1	0.590	0.042	55
f. Weed uneven	702.0	0.29	0.045	96
g. Sugar beet weed flat	699.2	-0.11	0.087	39
h. Sugar beet weed uneven	699.4	-0.089	0.039	89

TABLE II
RESULT OF DYNAMIC TESTS ON
SUGAR BEET WEED UNEVEN 700MM

Test	Measured distance y (mm)	Error (%)
1	698.9	-0.15
2	698.4	-0.23
3	699.0	-0.15
4	699.3	-0.098
5	698.9	-0.16
6	698.7	-0.19
7	699.1	-0.14
8	699.7	-0.046
9	699.7	-0.046
10	700.4	0.061

The tests on the eight tracks contains 70 image pairs each, resulting in a total distance of 700 mm for each test. The data are analyzed using a 51x51 pixel window corresponding to approximate 40x40 mm at height 450 mm. The error and variance of the measurement in driving direction of robot (y- direction) are presented in Table I.

In total 5.6 m on various surfaces using 560 stereo pairs, the variance is measured to 0.040 mm which means 99.7 % of the measurements are found within 0.60 mm. The variance measured includes both the manual positioning of the lathe carriage, and the error of the visual odometry system. Fig. 12 shows plots for the disparities measured and calculated traveled distance for the test on sugar beet sawed on uneven soil. The height dependence of the translation disparity (d_y) can clearly be seen by comparing it to the stereo disparity (d_{zx}).

The dynamic test is performed on one surface only, sugar beet with weed on uneven soil, Fig. 8h. The frame rate is set

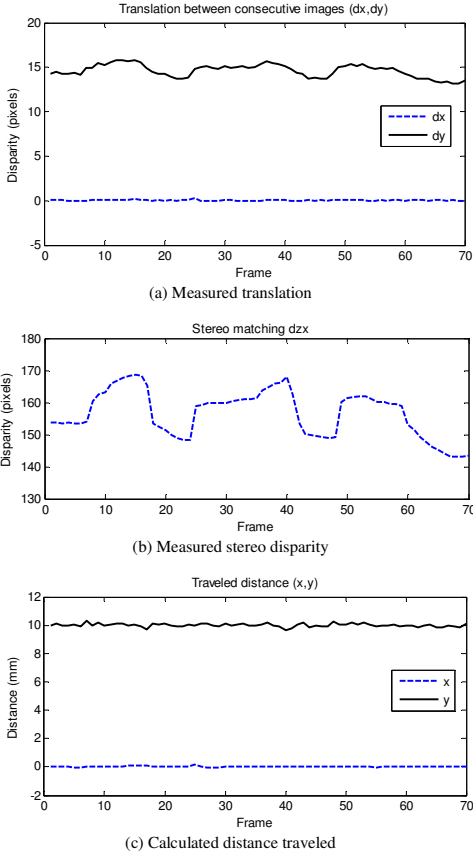


Figure 12. Example of measured translation, disparity and calculated distance. These plots are from the test on sugar beet on uneven soil.

to 5 Hz and the carriage is moved 10 times over the same distance with various speeds. The total distance for each test is 700 mm. The result is shown in Table II. Fig. 13 shows a plot of the disparity representing the height with respect to the estimated position for all 10 test runs.

An additional test is performed to evaluate the real-time performance. The algorithm is implemented in C++ using OpenCV library. The execution time per frame is measured and tested on the same data as in second test on all eight surfaces. This means that the images are loaded from the disk and the load time is included in this measurement. The measured mean time is 14 ms/frame. If the images are captured from the camera and then analyzed a frame rate of 30 images per second is obtained.

All previous tests have been performed without any barrel distortion compensation. The cameras are calibrated

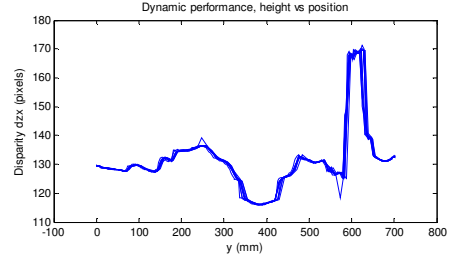


Figure 13. Result of 10 test run on sugar beet with weed on uneven ground.

using Camera Calibration toolbox for Matlab [15] and the images are compensated before they are analyzed. The result is shown in Table III. The execution time for analyzing each frame increases to 70ms/frame. The variances are similar to what obtained without compensation. For the test on “Sugar beet weed flat” increases the variance and the error due to some outliers. This result also affects the total variance for all test runs with compensation to 0.056 mm. Excluding that particular test run from the variance calculation it ends up at 0.030 mm.

IV. CONCLUSIONS

The system shows a small error, below 0.59% for all tests both with and without compensation for barrel distortion. The result is only slightly better for test with compensation. However, the compensation increases the calculation time, which means that the increased performance comes at the cost of frame rate.

The highest variance for a single run without compensation is found on the test “sugar beet weed flat”. It is measured to 0.087 mm. The accuracy is still sufficient for

TABLE III
COMPARISON BETWEEN COMPENSATED AND UNCOMPENSATED IMAGES ON 700MM

Test	Measured distance y (mm)	Variance (mm)	Measured distance y compensated (mm)	Variance compensated (mm)
a. Soil flat	702.5	0.0084	700.6	0.0075
b. Sugar beet flat	695.9	0.057	697.1	0.013
c. Sugar beet uneven	699.9	0.013	698.9	0.014
d. Sugar beet grass flat	702.1	0.024	700.6	0.021
e. Sugar beet grass uneven	704.1	0.042	703.9	0.057
f. Weed uneven	702.0	0.045	700.8	0.049
g. Sugar beet weed flat	699.2	0.087	702.8	0.23
h. Sugar beet weed uneven	699.4	0.039	697.7	0.046

crop-scale agricultural operations. The error for the dynamic test is less than 0.23% which indicates that the error depends more on the texture and height profile than of the speed.

The error and variance is very dependent on outliers, which usually occurs when the images contains shadows or the texture is poor. In these situations the similarity image contains several peaks, where the highest ones are candidates to be the selected match. Shadows can be reduced and texture can be enhanced by improving the illumination. The system is also sensitive to missing frames. However, these are both easy to detect and to compensate for, for instants by double the search area in next image.

Further work will include more real-time tests on a robot with steering control, using either row-following system or omnidirectional camera. The camera mounting and illumination has to be improved to reduce the number of outliers. Further more tests have to be done in real environment to evaluate the system on a robot.

REFERENCES

- [1] J.F. Reid, Q. Zhang, N. Noguchi, M. Dickson, Agricultural automatic guidance research in North America, *Computers and Electronics in Agriculture* 25, 2000, 155-167
- [2] T. Hauge, J.AMarchant, N.D. Tillet, Ground based sensing systems for autonomous agricultural vehicles, *Computers and Electronics in Agriculture* 25, 2000, 11-28
- [3] B. Åstrand, A.J. Baerveldt, Vision-based perception for an agricultural mobile robot for mechanical weed control, *Autonomous Robots* 13, 2002, 21-35.
- [4] S. Gan-Mor, R.L. Clark, B.L. Upchurch, Implement lateral position accuracy under RTK-GPS tractor guidance, *Computers and Electronics in Agriculture* 59, 2007, 31-38
- [5] M. Maimone, Y. Cheng, L. Mattheis, Two years of visual odometry on the mars exploration rovers, *Journal of Field Robotics* 24(3), 2007, 169-186
- [6] M. Agrawal, K.Konolige, Rough terrain Visual odometry, *Proceedings of the International Conference on Advanced Robotics (ICAR)*, Jeju island, South Korea, 2007
- [7] S. Ericson, B. Åstrand, Algorithms for Visual Odometry in Outdoor Field Environment, *Proceedings of 13th IASTED International Conference Robotics and Applications*, Würzburg, Germany, 2007, 287-292
- [8] W. Seemann, K.D. Kuhnert, Localisation for an outdoor robot by optically measuring the ground movement, *Proceedings of 13th IASTED International Conference Robotics and Applications*, Würzburg, Germany, 2007, 113-118
- [9] M. Zaman, High resolution relative localisation using two cameras, *Robotics and Autonomous Systems* 55, 2007, 685-692
- [10] P. Corke, D. Strelow, S. Singh, Omnidirectional visual odometry for a planetary rover. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sendai, Japan, 2004, 4007- 4012
- [11] E. Trucco, A. Verri, *Introductory techniques for 3-D computer vision*, (Prentice-Hall Inc. Upper Saddle River, New Jersey, USA)
- [12] Z. Zhu, T. Oskiper, O. Naroditsky, S. Samarasekera, H.S. Sawhney, R. Kumar, An Improved Stereo-based Visual Odometry System, *Performance Metrics for Intelligent Systems Workshop*, USA, 2006, 149-156
- [13] J. Westerweel, Effect on sensor geometry on the performance of PIV interrogation. *Proceedings of the 9th International Symposium on Applications of Laser Techniques to fluid Mechanics*, Lisbon, Portugal, 1998, 1-2
- [14] B. Jähne, *Digital image processing*. (Berlin: Springer, 2002).
- [15] J-Y Bouguet, Camera calibration toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/calib_doc

PAPER III

A VISION-GUIDED MOBILE ROBOT FOR PRECISION AGRICULTURE

S. Ericson and B. Åstrand (2009). "A vision-guided mobile robot for precision agriculture". In: Proceedings of 7th European Conference on Precision Agriculture. Wageningen, Netherland, pp.

623–630

A vision-guided mobile robot for precision agriculture

S. Ericson

*University of Skövde
Box 408, 54128 Skövde
Sweden
stefan.ericson@his.se*

B. Åstrand

*Halmstad University
Box 823, 30118 Halmstad
Sweden
bjorn.astrand@ide.hh.se*

Abstract

In this paper we have developed a mobile robot which is able to perform crop-scale operations using vision as only sensor. The system consists of a row-following system and a visual odometry system. The row following system captures images from a front looking camera on the robot and the crop rows are extracted using Hough transform. Both distance to the rows and heading angle is provided which both are used to control the steering. The visual odometry system uses two cameras in a stereo setup pointing perpendicular to the ground. This system measures the travelled distance by measuring the ground movement and compensate for height variation. Experiments are performed on an artificial field due to the season. The result shows that the visual odometry have accuracy better than 2.1% of travelled distance.

Keyword – visual odometry, row following

Introduction

The precision of the position determines what tasks an autonomous field robot can perform. High precision system can perform crop scale operations, while low precision systems can operate on a section of a field. Tasks like precision spraying or mechanical weed control require high precision of the measurement relative the crop. These systems can gain of using relative measurements instead of absolute. Since many agricultural fields are structured by rows, these can be used for navigation. When ploughing and harvesting you try to follow the line between treated and untreated area. In the case of crops that are sowed in rows like potatoes or sugar-beets you can follow these rows for weed control and harvesting.



Figure 1 Experimental platform, a modified Permobil exterior 90

The most common used sensor for mobile agricultural robots is a global navigation satellite system, GNSS (Lechner and Baumann 2000). A cheap GPS receiver provides a position accuracy of 30m. Enhanced GPS sensors based on carrier phase technology (RTK GPS) provides position accuracy of a few centimetres, at a high rate, up to 20Hz. However, it

requires a local reference sensor nearby the moving sensor. The RTK GPS sensor is used on commercially autopilot systems for tractors and harvesting machines. An evaluation of one of these systems shows a position accuracy of approximately 2.2 cm (Gan-Mor et al. 2007). A drawback of this sensor is that it suffers from drop outs. It also has a long setup time before it can deliver accurate position measurements. The GPS system provides a global position which can be used to know where on the field the robot is located. However, this position is of limited value for crop-scale operations, since the environment is not fixed and the position has to relate to a map created by previous generated map.

Another method is to use a computer vision system. The camera is a very powerful sensor, and the challenge is to extract the information of interest from the images. Most image processing requires a lot of computational power, but the rapid development of computers has made it possible to solve more advanced task. It has the potential to become a very cheap sensor in the future. This is shown in the cell phone development where one or two cameras can be found.

Today there are several autonomous agricultural vehicles that navigate using a row-following system. In (Åstrand and Baerveldt 2002) the authors use it for mechanical weed control of sugar-beet. They use a camera mounted in the front of the robot, where lines from the images are extracted using Hough transform and used in a control algorithm. The accuracy reported was down to 6 mm depending on crop size. Others (Hague and Tillett 2001) and (Tillett et al. 2002) use a filter approach to detect the crop rows, also for guiding an implement for mechanical weed control in sugar beet. They reported an accuracy of 16 mm. Further, systems for guidance have been reported where regression is used for the row detection (Billingsley and Schoenfisch 1996). The accuracy obtain was 20 mm. A similar approach is used by Sogaard and Olsen (2003), where a weighted linear regression is used to find rows, but without segmentation. The row position could be estimated to 6-12 mm, depending on crop size.

The Demeter system (Pilarski et al. 2002) is a speed rower using both RTK GPS and machine vision. The GPS system only provides position while the machine vision system has algorithms for shadow compensation, end of row detection and obstacle detection. Average tracking error from the GPS was 4-6cm and from the vision system 5-30 cm.

More recent implementation of a row-following algorithm has been done in (Leemans and Destain 2007), where it is used for controlling a precision seed drill. They use a modified Hough transform to detect previous saw rows in soil. The result reported was 23 mm. Another row detection system is presented in (Bakker et al. 2008). They also use Hough transform to detect rows of sugar beets with mean error at 22 mm.

This paper contributes with a method of navigating a mobile robot on an agricultural field using vision only. The objective with this paper is to show a method to provide a good local position relatively the crops. This position can be used for performing tasks as mechanical weeding and other operations close to the crops.

Material and method

The experimental platform

The mobile robot is developed based on an electrical wheel chair for outdoor use, a Permobil exterior 90, Figure 1. The Permobil is a four-wheeled mobile robot with electrical steering and back-wheel drive. It has a top speed of approximately 15 km/h and a claimed range of 30km (unmodified). It is modified with sensors, computers and a new control system for autonomous drive. A block diagram of the computer and sensor setup is shown in Figure 2.

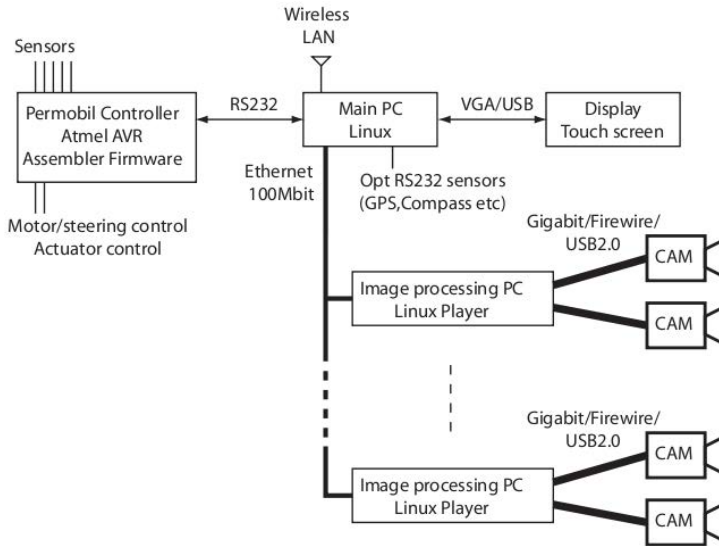


Figure 2. Hardware setup

There are three on-board computers connected in a network, a main computer running navigation software and two image processing computers. The main computer has integrated touch screen for monitoring and human interaction. One of the image processing computers is connector to a front looking cameras and the other computer is connected to two down-looking cameras. The image processing computers are equipped with an Intel Core2Duo 1,73 GHz and 3 Gb RAM. The main computer is a Roper 12-05ps. A control box (Permobil Controller, figure 2) has been build based on an Atmel AtMega128 to communicate between the Permobil hardware and the main computer. Further the robot has magnetical encoders mounted on the back driving wheel axis for position data and velocity control of the wheels. There is also space for optional sensors such as measurement of steering angle but this is not implemented yet. However, the Permobil provides a close loop control of steering angle, which is used.

There are three cameras on the robot, two for the visual odometry system and one for the row following system. The visual odometry cameras are placed behind the back wheel in a box pointing perpendicular to the ground. The row-following camera is mounted on a metal stand 1.4 meter above the ground, tilted 15° towards the ground. Figure 3 shows the modified Permobil. The camera used is a MV BlueFOX-120aC USB camera which can provide a frame rate of 100 frames/sec.



Figure 3. Modified Permobil during experiments.

All computers are running Ubuntu Linux 7.04 and Player (Gerkey et al. 2003). Custom drivers (server) have been developed for player that includes image acquisition, image processing and position calculation. These drivers are implemented using C++ and OpenCV (Bradski and Kaehler, 2008) to provide real time performance. All computers are running one server each and the main computer is also running a client. This client collects sensor data from all the servers and provides navigation control by setting the speed and steering references to the Permobil controller.

Robot control and sensor model

The robot is represented by an Ackermann model. Figure 4 shows the model of the robot on a field of structured rows. The measured properties are distance to row (s) and heading (α) from the row-following system and travelled distance from the visual odometry system. The steering angle (β) are controlled by method described in (Sotelo 2003).

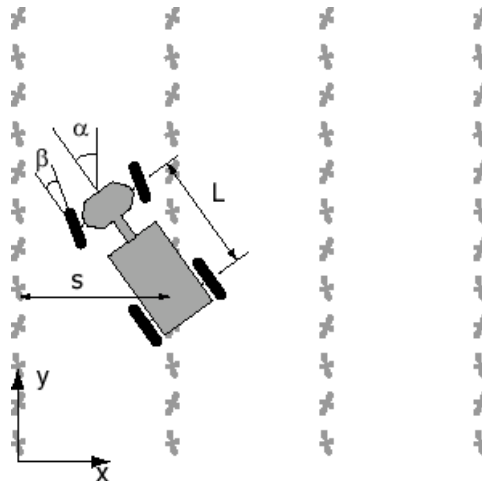


Figure 4. Robot model.

Visual Odometry

In (Ericson and Åstrand 2007) algorithms for visual odometry are evaluated. It shows that normalized correlation provides good result as long as there is texture enough to detect motion and maximum speed is not violated. This algorithm is developed further to deal with the uneven terrain expected on an agricultural field (Ericson and Åstrand 2008). It uses stereo cameras to calculate the height of the cameras. The travelled distance is measured by two consecutive frames and compensated for height variation. The error on a test track was less than 0.6% on 700mm distance.

This algorithm is implemented and runs on one of image processing computers. The speed of this computer vision system determines the maximum speed on the robot during test. This configuration provides a low speed system, but with the flexibility to change algorithm. The frame rate is 30 frames/second. Sharp images are provided by using a short exposure time (5 μ s). Controlled illumination is obtained with lamps connected to a dimmer. The triggering of the cameras is crucial to obtain the height information. A real time controller in one of the cameras is programmed to send a trig pulse to all cameras. In this way all cameras are triggered at the same time.

To increase the analyze speed a part of the image is acquired on the camera. Full resolution is 640x480, but only a window of 101x101 is extracted from both images where the matching window is the 51x51 pixels in the centre. In these tests the lens used gives approximately the resolution of 1 pixel/mm, and hence a 41mm x 41mm area is evaluated. It is important that this area has texture enough to provide a good match. By varying the clearance to the ground and processing speed, higher resolution can easily be obtained, at the cost of maximum speed.

Row-following

The row-following system uses Hough transform to detect lines on an agricultural field using the front looking camera (Åstrand and Baerveldt 2005). The captured images are converted to binary image by first extracting the colour of interest and then apply a threshold.

The threshold has to be adjusted so only the rows are detected. Only points on the ground are of interest so all points at and above the horizon can be ignored. The Hough transform is applied directly in the s - α plane to obtain the distance to the row (s) and heading angle (α). Valid lines have a heading within 10 degrees from the rows and distance within one meter. Two lines are detected and both lines are used as reference for the steering control. These values are used in the steering control.

Experiments and results

Experiments are performed on an artificial field to evaluate the performance of the sensors and the closed loop position controller. The field is simulated by placing red poker chips in two straight rows on a lawn. Grass is a representative texture for the visual odometry and the poker chips represent the crops. The reason for not using a real field is the lack of colour separation during the winter. The red colour is found by using Equation 1.

$$E = 2R - G - B \quad (1)$$

where E is a greyscale image
 R, G, B represents the colour channel red, green and blue

The robot is placed at the beginning of the rows and it is programmed to go autonomous a specified distance, in this case 10 meters. The steering is controlled only by row-following system, while the speed is set to a constant value. It is important to keep the speed below the maximum speed allowed for the visual odometry. When the visual odometry measures the end position the robot stops. The error is measured manually with a scale as error in row-following and error in odometry.

Figure 5 shows images from the row following system. Figure 5a shows the test field as seen from the image. Figure 5b is the binary image obtained after colour extraction and thresholding. Finally, Figure 5c shows the Hough space in s - α -plane. Each intersection represents one of the two lines. These are located at approximately $\alpha = 0^\circ$ and $s = -350$ mm and $s = 400$ mm. The actual row distance is 1 meter, and the difference is due to calibration error.

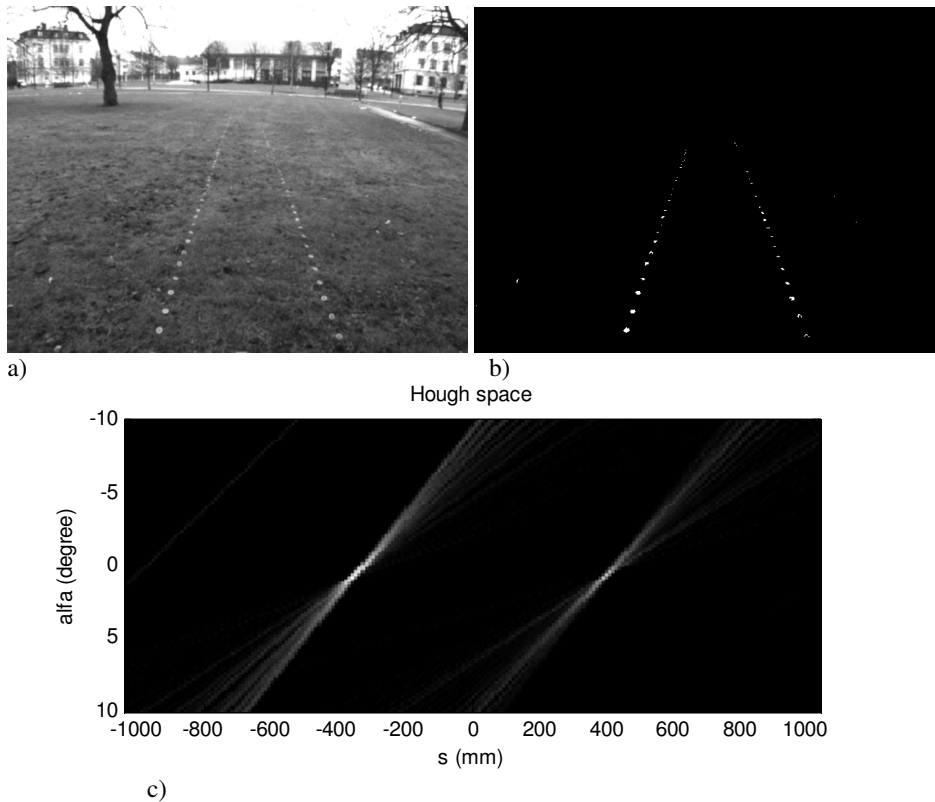


Figure 5. a) Captured colour image, b) Binary image after thresholding, c) Hough space in s - α -plane

The result of three different test runs on the same track is shown in Table 1. The measurement from the visual odometry is presented as measured distance along the row (y-direction). The result shows that the position is overestimated which can depend on calibration error.

Table I, Result of autonomous drive 10 m.

<i>Test</i>	<i>Measured distance by visual odometry (mm)</i>	<i>Measured position by scale y (%)</i>	<i>Error (%)</i>
1	11132	11280	1.3
2	10913	11150	2.1
3	10631	10550	0.7

A plot of the distance each frame measured by the visual odometry is shown in figure 6. The frame rate is 30 Hz and the figure shows that the speed is between 6-10 mm/frame. This corresponds to a speed of 0.18-0.30 m/s. The variation depends on the uneven terrain. The sum of all this measurements gives the total distance 10631 mm (test 3), and calculated error is 0.7%.

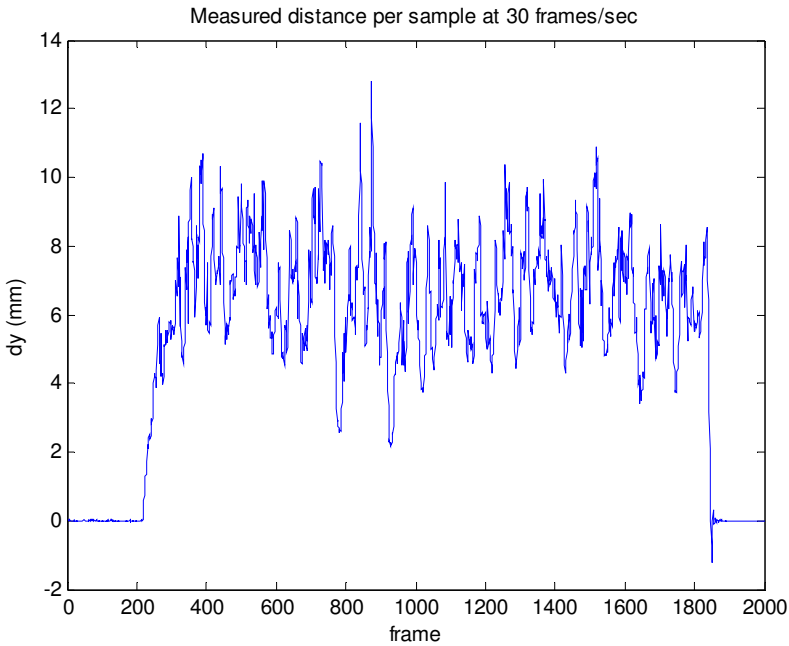


Figure 6. Measured distance from visual odometry system per frame, i. e. speed in y-direction.

Conclusions and further work

These experiments are first system trials of combining visual odometry with a row following system. The result shows that the precision of the system is promising. During these test the distance was overestimated with 0.7% and 2.1% error. This is enough for crop scale operations. Further work is to do more experiments on real fields. A more extensive calibration has to be done, and error variance has to be measured. Then these values can be used in a Kalman filter, where data from other available sensors can contribute to a better position estimation.

References

- Anonymous, Permobil, <http://www.permobil.com/> (Accessed 2009-01-31)
- Bakker, T., Wouters, H., Asselt, K., Bontsema, J., Tang, L., Müller, J. and Straten, G., 2008. A vision based row detection system for sugar beet, *Computers and Electronics in Agriculture*, 60:87-95
- Billingsley, J. and Schoenfisch, M., 1996. The successful development of a vision guidance system for agriculture, *Computers and Electronics in Agriculture* 16:147-163
- Bradski, G. and Kaehler, 2008. *A. Learning OpenCV*, O'Really Media Inc., USA, pp555
- Ericson, S. and Åstrand, B., 2007. Algorithms for Visual Odometry in Outdoor Field Environment, In: *Proceedings of 13th IASTED International Conference Robotics and Applications*, Würzburg, Germany, pp 287-292
- Gan-Mor, S., Clark, R. L. and Upchurch, B. L., 2007. Implement lateral position accuracy under RTK-GPS tractor guidance. *Computers and Electronics in Agriculture* 59:31-38
- Gerkey, B., Vaughan, R. T. and Howard, A., 2003. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In: *Proceedings of the 11th International Conference on Advanced Robotics*, Coimbra, Portugal, pp 317-323
- Hague, T. and Tillett, N.D., 2001. A bandpass filter approach to crop row location and tracking. *Mechatronics* 11:1-12
- Lechner, W. and Baumann, S., 2000. Global navigation satellite systems. *Computers and Electronics in Agriculture* 25:67-85.
- Leemans, V. and Destain, M.-F., 2007. A computer-vision based precision seed drill guidance assistance, *Computers and Electronics in Agriculture* 59:1-12
- Pilarski, T., Happold, M., Pangels, H., Ollis, M., Fitzpatrick, K. and Stentz, A., 2002. The Demeter System for Automated Harvesting, *Autonomous Robots* 13:9-20
- Sogaard, H.T. and Olsen, H.J., 2003. Determination of crop rows by image analysis without segmentation, *Computers and Electronics in Agriculture* 38:141-158
- Sotelo, M. A., 2003. Lateral control strategy for autonomous steering of Ackerman-like vehicles, *Robotics and Autonomous Systems* 45:223-233
- Tillett, N. D., Hague, T. and Miles, S. J., 2002. Inter-row vision guidance for mechanical weed control in sugar beet, *Computers and Electronics in Agriculture* 33:163-177.
- Åstrand, B. and Baerveldt, A.J., 2002. Vision-based perception for an agricultural mobile robot for mechanical weed control, *Autonomous Robots* 13:21-35.
- Åstrand, B. and Baerveldt, A.J., 2005. A vision based row-following system for agricultural field machinery, *Mechatronics* 15:251-269.

PAPER IV

ROW-DETECTION ON AN AGRICULTURAL FIELD USING OMNIDIRECTIONAL CAMERA

S. Ericson and B. Åstrand (2010). "Row-detection on an agricultural field using omnidirectional camera." In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010. IROS 2010, pp. 4982–4987. ©2010, IEEE. Reprinted, with permission, from S. Ericson & B. Åstrand

Row-detection on an agricultural field using omnidirectional camera.

Stefan Ericson and Björn Åstrand

Abstract— This paper describes a method of detecting parallel rows on an agricultural field using an omnidirectional camera. The method works both on cameras with a fisheye lens and cameras with a catadioptric lens. A combination of an edge based method and a Hough transform method is suggested to find the rows. The vanishing point of several parallel rows is estimated using a second Hough transform. The method is evaluated on synthetic images generated with calibration data from real lenses. Scenes with several rows are produced, where each plant is positioned with a specified error. Experiments are performed on these synthetic images and on real field images. The result shows that good accuracy is obtained on the vanishing point once it is detected correctly. Further it shows that the edge based method works best when the rows consists of solid lines, and the Hough method works best when the rows consists of individual plants. The experiments also show that the combined method provides better detection than using the methods separately.

I. INTRODUCTION

Mobile robots for use on an agricultural field require reliable sensors for both localization and perception. Today, one of the most common used sensors for agricultural machinery is the RTK-GPS. It is mainly used as position measurement for tractor autopilots, where the farmer can drive in straight rows with minimum of overlap between rows. It is also used on autonomous agricultural robots but only for research. The drawbacks of the RTK-GPS are the dropouts and the requirement of clear view of the sky. The cost has also mentioned as an issue, but promising work shows a way of building low-cost RTK-GPS using open source library [1].

The use of cameras for localization has been seen more as a complementary method to GPS. However, advantages of using camera are that it can be used for simultaneously localization and mapping (SLAM) and obstacle avoidance. It could also provide low-cost system. In the case of navigation on an agricultural structured field, one of the most important tasks is to keep track of the rows and to separate the crops from weed and soil. Several agricultural robots has been presented in the literature, some navigating using only GPS [2], others with vision [3], [4], [5], and some with sensor fusion between several sensors [6]. In [7] a mobile robot for automated weed control is presented where perspective cameras are the main sensor. In own previous work [8] a mobile robot was presented using row-following from perspective camera and visual odometry for navigation. Fig.

1 shows the mobile robot on a row-structured agricultural field.



Fig. 1: Mobile experimental robot on a row-structured agricultural field

The vision guided robots introduced so far have used perspective cameras. There are several advantages of using an omnidirectional camera instead. First more and longer rows can be captured, i.e. more robust to weed pressure and missing crops and rows. Second, it sees plants beside the robot which may give better estimate of alignment error. Finally it sees behind the robot which gives the opportunity to achieve better row-guidance at end of row, and enables monitoring of field operations. Among the omnidirectional cameras there are both cameras with catadioptric lens and with fisheye lens. The major difference is the range of azimuthal view. A fisheye lens starts from zero which means it sees straight ahead, and end somewhere above 90° . The catadioptric lens on the other side cannot see straight ahead due to its construction, but it has a wider range above 90° . The image analysis on omnidirectional images can be categorized in two groups, those who require the image to be unwrapped and those who are applied directly on the omnidirectional image. The unwrapping is a time consuming process and for real-time applications on a mobile robot the latter is to prefer. The algorithms used in this work do not need the images to be unwrapped.

Successful work on omnidirectional images has recently been presented in [9] where lines vertical to the camera is extracted. A SIFT-like descriptor is used for matching and tracking these lines between frames. This method is used for localization of a mobile robot [10], and it provides accurate heading information as well as translation. A drawback of the system is that it does not deal with tilt. In an agricultural

S. Ericson is with School of Technology and Society, University of Skövde, Skövde, Sweden stefan.ericson@his.se

B. Åstrand is with the School of Information Science, Computer and Electrical Engineering, Halmstad University, Halmstad, Sweden, bjorn.astrand@hh.se

scene the robot is moving in uneven terrain and the tilt is required to estimate the position of the row. Further more only a few lines can be expected to be radial to the camera on a field consisting of parallel rows. At the beginning of a row there will be one line ahead of the robot and while driving along a row, radial lines can be found in front and behind the robot. This is too few lines and hence this method is not suitable for this application.

In [11] the authors present a novel edge-based method to find lines and further rectangles in catadioptric images. The lines are extracted using edge detection on a grayscale image. All connected edge points are grouped, and the endpoints are found. Lines shorter than a specified threshold are rejected. The authors use a method where the points are mapped via a sphere. In that way all straight lines in the omnidirectional image can be represented by a great circle on the sphere. Each line is then splitted so all points on the same line can be projected on one great circle. The last step in the algorithm is to merge all lines which can be represented by the same great circle. This method is applied in finding the attitude of an UAV [12] and to navigate in urban environment by finding and tracking the vanishing point [13]. However, this method requires well defined edges, consisting of connected edge points in the direction of the lines.

In an agricultural scene the rows consist of single plants placed on a line with different distance. Plants close to the robot, or in this case close to the camera, hold important information about the robot's position relative the row. In this particular area the rows may not be viewed as solid, rather as individual plants. The method presented in [11] uses edge detection such as Canny for extracting line. This method is expected to work in areas far away from the robot, where the rows will be seen as solid lines. In the areas close to the robot, the rows may only consist of the shapes of the individual plants. In this area the Hough transform is potentially better to use. A method for detecting lines in catadioptric images using Hough transform is presented in [14].

This paper contributes with a method of detecting parallel rows on an agricultural field using an omnidirectional camera. The rows can consist of either solid lines or individual plants. Both catadioptric and fisheye lens are supported, and the algorithm extracts the vanishing point which contains information about both heading and tilt. This work evaluates the two methods to find lines on agricultural scenes and suggests a method that combines the Hough transform and the edge method.

II. METHOD

This work differs from the work done in [11] in three ways. First, the calibration of the omnidirectional camera uses the Taylor model [15] which allows both catadioptric and fisheye lenses. Second, a Hough transform step is added to find rows consisting of individual plants. Finally a second Hough transform is applied to find the vanishing point of the parallel row structure.

A. Camera calibration

It is assumed that the camera uses a lens/mirror combination that provides a single effective viewpoint, which means that all incoming rays can be modeled to intersect the center point of a unit sphere. Then each ray can be represented by a unit vector on this sphere as shown in Fig. 2. Further it is assumed that the lenses and mirrors are symmetric and that the sensor plane is perpendicular to the optical axis. In [16] it is shown that both fisheye and catadioptric cameras can be projected onto the image plane using two functions g and h as shown in (1).

$$\alpha'' \mathbf{p}'' = \begin{bmatrix} h(\|\mathbf{u}''\|) \mathbf{u}'' \\ g(\|\mathbf{u}''\|) \end{bmatrix} = \mathbf{P}'' \mathbf{X} \quad (1)$$

where $\alpha'' > 0$ is factor, $\mathbf{u}''$ is the point on sensor plane, $\mathbf{P}'' \in \mathbb{R}^{3 \times 4}$ is projection matrix and $\mathbf{X}$ is the scene point. In [15] a unified method to calibrate both fisheye and catadioptric camera is presented. This is done by rewriting the two functions (g and h) as one function g/h . By representing this function with a Taylor approximation, the relation between a scene point and a point in the sensor plane can be expressed as shown in (2).

$$\alpha'' \mathbf{p}'' = \begin{bmatrix} \mathbf{u}'' \\ a_0 + a_2 \|\mathbf{u}''\|^2 + \dots + a_N \|\mathbf{u}''\|^N \end{bmatrix} = \mathbf{P}'' \mathbf{X} \quad (2)$$

where a_n is coefficients of the Taylor polynomials. The advantage of using this representation is that it is valid for both catadioptric and fisheye lenses. Hence the method can be used for both types. The Matlab toolbox described in [17] is used for the calibration procedure, which provides the relation between an image point in pixels with a unit vector from the single effective view point.

B. Row model

Rows in the scene can be represented by a plane cutting the unit sphere in the camera model, i.e. each point on the row can be expressed as a vector lying both on the plane and the sphere. The plane can further be expressed by its normal, which means each row can be expressed as this normal. From here on normal is used for denoting the normal vector perpendicular to the plane. Fig. 2 shows how a row of plants is projected on the sensor plane and the corresponding image.

C. Edge method

Crop rows far away from the camera will be viewed as solid lines. Hence they can be modeled as lines which can be detected with an edge method. This part is similar to the method described in [11]. The first step is to apply the Canny algorithm to the image. Then a mask is applied to remove points not of interest. It can be structures from camera mounting or points close to the edge of visible field. Further each line is divided into sections where each section can be represented by one normal vector on the unit sphere. That means only points in the same direction are kept in one

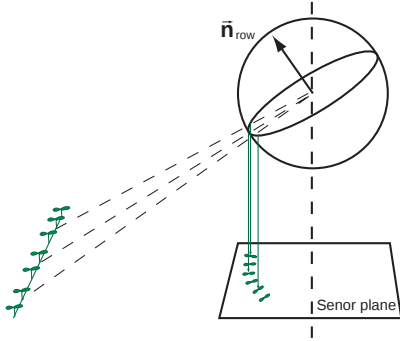


Fig. 2: Projection through the lens on to the sensor plane by a unit sphere model.

line. Minimum line length is specified and line shorter than this threshold is removed. The last step is a merging step that merges lines describing the same row. The orthogonal distance between all normals is calculated. Normals close to each other, i.e. the distance is shorter than a specified threshold, are considered to be the same normal. The new normal is estimated by using singular value decomposition of all points.

D. Hough method

The Hough method consists of a Hough transform followed by a non linear optimization step. The original image is used as input. Hence a separate mask can be used to specify the area of interest. First a Gaussian smooth filter is applied to reduce noise in the image. Then an adaptive threshold is applied to produce a binary image, where corresponding unit vectors can be calculated for each point. The threshold is tuned to allow a certain number of points to be selected. In this way the computation time of the Hough transform can be controlled. The parameters selected for the Hough transform is spherical coordinates of the normal to the plane representing the rows.

$$\vec{n}_r = [\sin(\varphi_r)\cos(\theta_r) \quad \sin(\varphi_r)\sin(\theta_r) \quad \cos(\varphi_r)]^T \quad (3)$$

where $0 < \varphi_r < \pi$ and $0 < \theta_r < 2\pi$. Let $\vec{n}_r = [n_x \quad n_y \quad n_z]^T$ denote the normal to the plane. Then the points on a row will satisfy (4).

$$n_x \cdot \sin(\varphi)\cos(\theta) + n_y \cdot \sin(\varphi)\sin(\theta) + n_z \cdot \cos(\varphi) = 0 \quad (4)$$

The size of the accumulator is kept small to increase the processing speed. Experiments in this paper uses accumulator of size 90×90 . The accuracy of the normal describing the plane is increased by applying a line fitting step as suggested in [14].

E. Finding vanishing point

The vanishing point contains information about the heading of the robot and also tilt. Similar to how points on a

row produce vectors that lies on a plane, several parallel rows will produce normals that define a plane. Hence the same method can be used to find the major direction of all rows. The normal to the multiple planes will point out the vanishing point in the image. The method used is the same as the Hough transform used for finding normals representing the rows. The number of rows is generally low which means the second Hough transform will execute fast. The range of φ and θ is selected to correspond to the range of expected vanishing points. For the case with a mobile robot in an agricultural field, the vanishing point is expected to be along the horizon. Hence points too high in the sky are rejected.

The output from the second Hough transform is used as input in a non linear optimization which gives the resulting vector describing the vanishing point. Fig. 3 shows the vector describing vanishing point, vectors describing the rows and finally each individual point that is seen in the image plane.

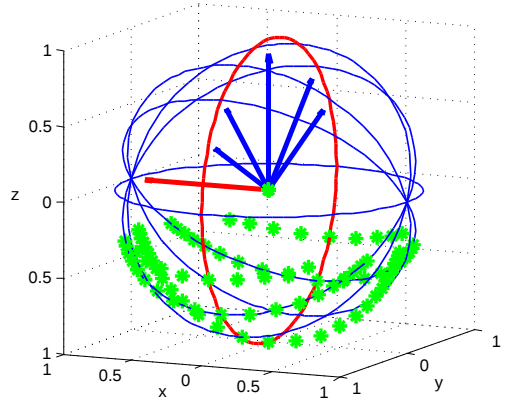


Fig. 3: Normals and planes in unit sphere describing vanishing point(red) and rows (blue). Projections of individual plants are marked in green.

F. Combined method

The combined method takes advantage of both edge method and the Hough method by feeding the second Hough transform with normals from both methods. These normals are weighted according to how many points on the row that have contributed to the normal. This leads to higher weight for rows that were built from many points. For the edge method this corresponds to longer connected edges, and for Hough method the value in the accumulator, i.e number of points contributing to the line. This step reduces the use of the merging step in the edge method since two short lines on the same great circle contributes to the same cell in the second Hough transform. To adjust the balance between the methods, a general weight w is applied, where $0 < w < 1$.

Then the normals are weighted according to (5).

$$\vec{n}_{comb} = \begin{cases} w \cdot \vec{n}_{edge} & \text{for edge normals} \\ (1 - w) \cdot \vec{n}_{hough} & \text{for Hough normals} \end{cases} \quad (5)$$

The weight w is determined by experiments and in this paper 0.7 is used.

III. EXPERIMENTS

Experiments are performed on synthetic images to evaluate algorithm performance and on real field images to validate the result.

A. Hardware setup

The camera used for the experiment is a Prosilica GC2450C, which provides images with 2540x2040 pixels. It is connected using Gigabit Ethernet, and lens is of c-mount type. Two different lenses are used, one fisheye and one catadioptric. The fisheye lens is a FE185C057HA from fujinon, and the catadioptric is a PAL-S25G3817-27C from American Accurate Components, Inc. The camera is mounted on a stand with the camera pointing downwards. The camera and the two lenses are calibrated using multiple chessboard images, and the calibration data is stored for use in the generation of synthetic images.

B. Synthetic images

The synthetic images are generated by projecting rows onto the sphere. The rows are specified by the distance between rows and the distance between each individual plant. The position of a plant is then projected onto the sensor plane by using real calibration data from different lenses. A plant is build of a model with four leaves. All plants in the image are similar, i.e no scaling is applied. The camera position is defined by setting the yaw, pitch and roll relative the plane with the rows. From these two orthogonal vanishing point is calculated, one along the rows, and one perpendicular to the rows. Hence a true value for the vanishing point exists, which will be used as reference to the estimated vanishing point. Further, background noise is added to the image and a circular mask is applied. Fig. 4 shows two synthetic images with random yaw and pitch. One use calibration data from a fisheye camera and the other from a camera with catadioptric lens. The level of the background noise is 0.2 and 0.5.

There are three types of noise that can be added to the images. First, there is background noise as mentioned above, which is created by adding salt and pepper noise followed by Gaussian filtering. Second there is a noise on the position of each individual plant. It is introduced as an error in plant distance or row distance. The latter affects how straight the rows are. The last type of noise is created by adding plants at random position. This corresponds to weed in agricultural images.

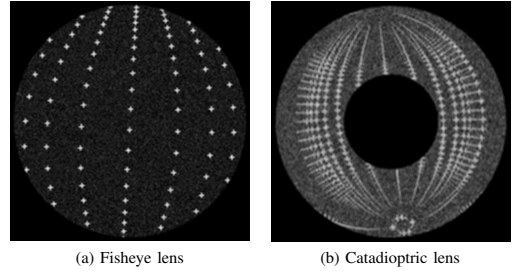


Fig. 4: Synthetic images with different camera tilt and background noise level.

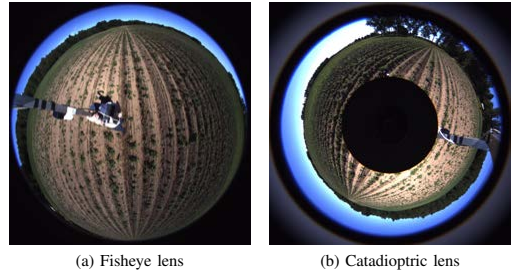


Fig. 5: Images of row structure on real field.

C. Evaluation on real field images

The methods are validated by using images from a real field. Typical images are shown in Fig. 5. The green parts of the image is used. The main difference from the synthetic images is that these images have visible areas above the field. This area has to be removed before the analysis, which is done by tracing the boundary of the regions and then calculate a mask from the result.

IV. RESULTS

A. Evaluation on synthetic images

Rows are detected in each image using both Edge and Hough method. Fig. 6 and Fig. 7 show the result for a fisheye image and an omnidirectional image. Rows detected by the Edge method are marked in blue and by Hough method in red. It can be seen that the Hough method captures the center of the row while the edge method detects the two edge lines of the row, i.e. row boundary.

First test evaluates the accuracy of estimated vanishing point by calculating the angle between true value and the estimated. This test is done on 100 randomly generated images, where yaw, pitch and lens type are varied. Hence both on fisheye and catadioptric images are used. The only noise added is background noise with density 0.1. The plant distance is low so almost solid lines are generated. The result is shown in Table I. It shows that the mean error is very

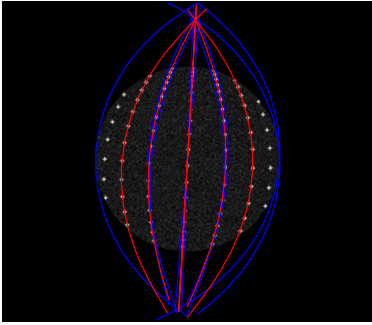


Fig. 6: Rows found in fisheye image. Edge (blue), Hough (red).

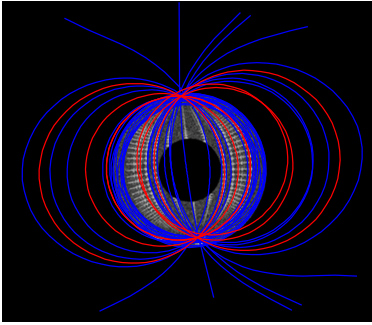


Fig. 7: Rows found in catadioptric image. Edge (blue), Hough (red).

small for correct found rows, so further experiments focus on detection rate.

The second test evaluates sensitivity to different plant distance. The detection rate is calculated as the quote between the number of correct rows found and total number of rows found. The result is presented in Fig. 8 for fisheye image and in Fig. 9 for omni image.

The notable result is that there is a major difference between the different lens types. For fisheye lens, the best result is obtained using the Hough transform, but for the catadioptric lens the Edge method works best. This is due to the shape of the rows in the images. It can be seen in the images that a catadioptric lens captures more points close to the vanishing point, and hence the rows are shown as solid lines. For the fisheye lens the rows are better described as

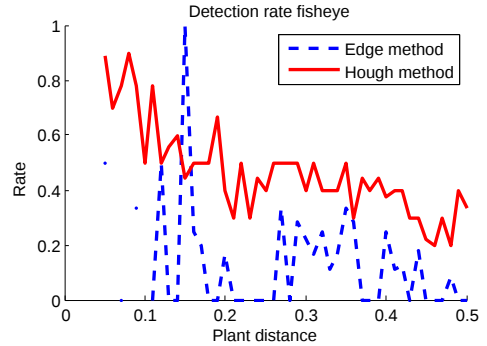


Fig. 8: Detection rate for fisheye image

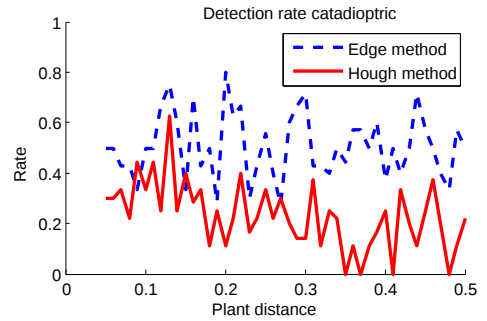


Fig. 9: Detection rate for omni image

individual plants, and hence Hough transform is better.

The last test on the synthetic images compares the two individual methods to the combined method. Images are generated randomly with different noise levels, lens types and plant distance. Each image is evaluated using all three methods. The result is presented as pass or fail, where it is classified as pass if the estimated vanishing point is within a certain distance from the reference. The amount of noise added is high which gives a lot of failures. Table II shows the result.

TABLE I: Accuracy of estimated vanishing point.

	Mean error (10^{-3} rad)	Max error (10^{-3} rad)	Variance 10^{-8}
Edge method	0.12	1.8	7.9
Hough method	0.036	0.17	0.13
Combined method	0.029	0.35	0.23

TABLE II: Comparison between methods.

	Edge method (failures)	Hough method (failures)	Combined method (failures)
Fisheye lens	30	18	15
Omni lens	27	48	23
Sum	57	66	38

B. Evaluation on real field images

The result from the test on real field images is shown in Fig. 10 and Fig. 11. Rows are found with both edge and Hough method in the fisheye image. In the catadioptric image no edge lines are found. The Hough transform provides two candidates of vanishing point, where one is aligned with the row structure, and the other is rejected. These images also shows that there is a small error in the calibration since the curvature of the lines is too small.

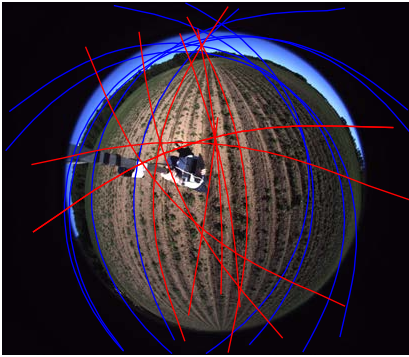


Fig. 10: Lines found in fisheye image. Edge (blue), Hough (red).

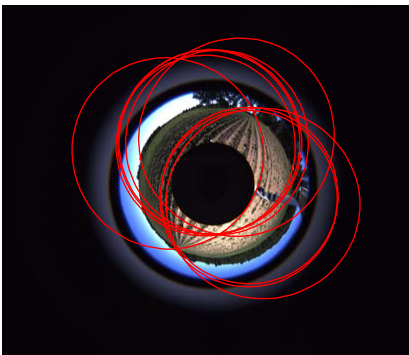


Fig. 11: Line found in catadioptric image. Edge (blue), Hough (red).

V. CONCLUSIONS AND FUTURE WORKS

In this paper we have evaluated two methods of extracting rows from an omnidirectional camera, and we have suggested an own combined method. The result from tests on synthetic images shows that the vanishing point can be estimated with high accuracy. It also shows that the detection rate varies between the methods and the lens used. The edge based method works best when the rows consist of solid lines, and the Hough method works best when the rows consists of individual plants. Hence, the two methods will work differently

on images from fisheye respectively catadioptric lenses. The result from the combined method shows that better detection is obtained than using the methods separately.

Further work is to evaluate different resolutions of images, and to transform the calibration data to the new resolution. The algorithm will be used for positioning a mobile robot on an agricultural field. That would require a real time implementation of the code and a control system of the steering.

REFERENCES

- [1] T. Takasu and A. Yasuda, "Development of the low-cost rtk-gps receiver with an open source program package rtklib," in *International Symposium on GPS/GNSS, International Convention Center Jeju, Korea*, November 4-6 2009.
- [2] A. Stoll and H. Dieter Kutzbach, "Guidance of a forage harvester with gps," *Precision Agriculture*, vol. 2, no. 3, pp. 281–291, 2000.
- [3] J. Billingsley and M. Schoenfish, "The successful development of a vision guidance system for agriculture," *Computers and Electronics in Agriculture*, vol. 16, no. 2, pp. 147 – 163, 1997, robotics in Agriculture.
- [4] J. Reid and S. Searcy, "Vision-based guidance of an agriculture tractor," *Control Systems Magazine, IEEE*, vol. 7, no. 2, pp. 39 – 43, apr 1987.
- [5] N. D. Tillett, T. Hague, and S. J. Miles, "Inter-row vision guidance for mechanical weed control in sugar beet," *Computers and Electronics in Agriculture*, vol. 33, no. 3, pp. 163 – 177, 2002.
- [6] T. Bakker, H. Wouters, K. van Asselt, J. Bontsema, L. Tang, J. Müller, and G. van Straten, "A vision based row detection system for sugar beet," *Computers and Electronics in Agriculture*, vol. 60, no. 1, pp. 87 – 95, 2008.
- [7] B. Åstrand and A.-J. Baerveldt, "A vision based row-following system for agricultural field machinery," *Mechatronics*, vol. 15, no. 2, pp. 251 – 269, 2005.
- [8] S. Ericson and B. Åstrand, "A vision-guided mobile robot for precision agriculture," in *Proceedings of 7th European Conference on Precision Agriculture*, Wageningen, Nederland, July 7-9 2009, pp. 623–630.
- [9] D. Scaramuzza, N. Cribble, A. Martinelli, and R. Siegwart, "Robust feature extraction and matching for omnidirectional images," in *Field and Service Robotics: Results of the 6 th International Conference (STAR: Springer Tracts in Advanced Robotics Series Volume 42)*, vol. 42. Springer, 2008, pp. 71–81.
- [10] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Transactions on Robotics, Special Issue on Visual SLAM*, vol. 24, no. 5, 2008.
- [11] J. Bazin, I. Kweon, C. Démonceaux, and P. Vasseur, "Rectangle extraction in catadioptric images," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2007, pp. 1–7.
- [12] C. Démonceaux, P. Vasseur, and C. Pegard, "Uav attitude computation by omnidirectional vision in urban environment," in *2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 2017–2022.
- [13] J. Bazin, I. Kweon, C. Démonceaux, and P. Vasseur, "A robust top-down approach for rotation estimation and vanishing points extraction by catadioptric vision in urban environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, 2008, pp. 346–353.
- [14] X. Ying and Z. Hu, "Catadioptric line features detection using hough transform," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4, 2004.
- [15] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *Proceedings of IEEE International Conference on Computer Vision Systems*, vol. 1. Citeseer, 2006, pp. 45–52.
- [16] B. Micušik, "Two-view geometry of omnidirectional cameras," Ph.D. dissertation, Czech Technical University, 2004.
- [17] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *Proc. of the IEEE International Conference on Intelligent Systems, IROS06, Beijing, China*. Citeseer, 2006.

PAPER V

EGO-MOTION ESTIMATION BY AN AGRICULTURAL FIELD ROBOT USING VISUAL ODOMETRY

S. Ericson and B. Åstrand (2016a). "Ego-motion estimation by an agricultural field robot using visual odometry". Submitted to Biosystems Engineering

Ego-motion estimation by an agricultural field robot using visual odometry

Stefan Ericson^{a,*}, Björn Åstrand^b

^a*School of Engineering Science, University of Skövde, Skövde, Sweden*

^b*School of Information Science, Computer and Electrical Engineering, Halmstad University, Halmstad, Sweden*

Abstract

This paper analyses two visual odometry systems for use in an agricultural field environment. The impacts of various design parameters and camera setups are evaluated in a simulation environment. Four real field experiments were conducted using a mobile robot operating in an agricultural field. The robot was controlled to travel in a regular back-and-forth pattern with headland turns. The experimental runs were 1.8–3.1 km long and consisted of 32–63,000 frames. The results indicate that high-resolution images captured from forward-facing stereo cameras mounted with a large baseline will give the best results. The algorithm must be able to reduce error accumulation by adapting the frame rate to minimize error. The results also illustrate the difficulties of estimating roll and pitch using a downward-facing camera. The best results for full 6-DOF position estimation were obtained on a 1.8-km run using 6680 frames captured from the forward-facing cameras. The translation error (x, y, z) is 3.76% and the rotational error (i.e., roll, pitch, and yaw) is 0.0482 deg m⁻¹. The main contributions of this paper are an analysis of design option impacts on visual odometry results and a comparison of two state-of-the-art visual odometry algorithms, applied to agricultural field data.

Keywords: Visual odometry, Agricultural field robots, Visual navigation

1. Introduction

Visual odometry (VO), a term introduced by (Nistér et al., 2004), is a method for estimating the position of a camera from an image sequence. In VO, consecutive image frames in a sequence are matched for correspondence and the relative poses between the frames are accumulated. This estimates the travelled path with up to six degrees of freedom (DOF). This technique is applied to agricultural field robots to increase navigation precision compared with that of current GPS navigation systems and to make robots that can operate closer to crops than can current systems.

The challenge of developing a vision system for an autonomous agricultural robot is to achieve reliability and robustness. Any solution must handle common situations such as varied weather and light conditions and interfering moving objects. The use

of VO in agricultural fields has been presented in various papers addressing the system level, but the results are rarely presented so as to make it easy to understand how different design options contribute to the final results. Compared with many other papers, this paper presents results in greater detail to clarify the effects of different design choices.

To improve the accuracy of VO, this paper seeks new knowledge of cumulative error when VO is used in an agricultural field environment. The accuracy is evaluated by comparing algorithms with different parameter settings. Experiments are conducted on both simulated data and data captured from a mobile robot, and both urban and agricultural environments are studied. The main contributions of this paper are to analyse the impact of various design options on visual odometry results and to compare two state-of-the-art visual odometry algorithms applied to agricultural field data.

*Corresponding author. School of Engineering Science. Box 428, 54128 Skövde, Sweden. Tel: +46 500 448509

Email address: stefan.ericson@his.se (Stefan Ericson)

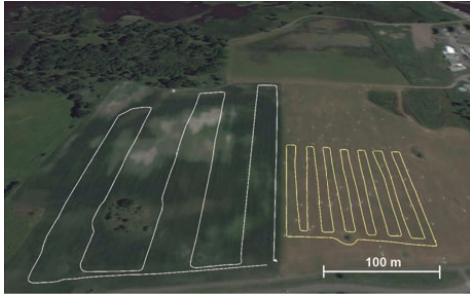


Figure 1: The test field showing GPS tracks of two test runs on Google Earth. Map data: Google, Lantmäteriet, Metria. The left-hand track represents the Field 1 experiment and the right-hand track the Field 3 experiment.

2. Related work

This section describes the related work from a general perspective with a focus on agricultural applications and on how to benchmark the results. Related work according to the various steps of the VO pipeline is explored in Section 3.

VO has been around for many years, one of the pioneering studies being by Nistér et al. (2004). Their work introduces an algorithm in which feature points are extracted from the images, matched to each other, and finally used for motion estimation. Outliers among the points are removed using random sample consensus (RANSAC) (Nistér, 2005). The algorithm is applied to a dataset acquired using a vehicle with a forward-facing stereo camera. Many state-of-the-art methods still use the same approach (Kitt et al., 2010; Geiger et al., 2011; Cvišić & Petrović, 2015) but differ slightly in how the features are extracted and matched and in how the outliers are removed. Position estimation can also be improved by making assumptions as to the environment or by adding a motion model of the vehicle, the goal being to reduce the cumulative error.

There are publicly available datasets for VO evaluation (Geiger et al., 2012). One such dataset, the KITTI Vision Benchmark Suite, consists of several sequences of images captured from forward-facing cameras in an urban environment. Ground truths are available for some, but not all, sequences; the remaining sequences are used for the evaluation of

algorithms and a ranking is published online¹.

Similar datasets are unavailable for the agricultural case, so a similar comparative ranking does not exist. One reportedly successful method using downward-facing cameras in an agricultural application has been presented by Jiang et al. (2014). Their robot, Gantry, is in the form of a three-meter-high square-shaped table with a combined driving and steering wheel at each leg. Two downward-facing cameras mounted at the top of the robot are used for the VO. In one experiment conducted in a soybean field, the path follows a regular back-and-forth track with a total of 13 headland turns. This experiment has a track length of 2.5 km and consists of 11,700 frames. Their results indicate that the translation error (2-DOF) was under 5.12 m, which corresponds to 0.2% of the travelled distance. A shorter path (i.e., 386 m and 1,300 frames) on a grass road was also evaluated and the reported result is 1.6%.

Piyathilaka & Munasinghe (2011) did not achieve the same accuracy when using VO on agricultural fields. Their approach was to use a monocular camera and a vehicle model to estimate the position in 3 DOF. The error of their VO system was 17% on a 35-m back-and-forth track. Combining the monocular camera system with an external camera system improved the accuracy, resulting in only 2.8% error. We previously achieved 0.6% error in 2-DOF estimation (Ericson & Åstrand, 2008) in a laboratory experiment using downward-facing stereo cameras.

The performance of previous related work is summarized in Table 1. The various methods mentioned in the table are further explained in Section 3.

Two state-of-the-art methods are selected for evaluation in this paper, the method used with the Gantry robot (Jiang et al., 2014) and the C++ VO library Libviso (Geiger, 2015; Geiger et al., 2011). Gantry was specifically developed for use in agricultural fields and its reported error was under 0.2%. The Libviso method is also intended for use in the agriculture field environment (Kameras als Augen des Jät-Roboters). The highest-ranked VO algorithm on the KITTI benchmark (Cvišić & Petrović, 2015) is based on the Libviso method. Cvišić & Petrović (2015) use the same feature extractor but apply a more sophisticated outlier rejector, so only

¹The KITTI Vision Benchmark Suite, http://www.cvlibs.net/datasets/KITTI/eval_odometry.php; accessed 20 June 2016.

Table 1: Summary of related work using VO in an outdoor environment. "DOF" describes the degrees of freedom. "Method" specifies the motion estimation method, whether the camera is mono or stereo, and the direction the camera faces, i.e., forward (FW) or downward (DW). See explanation of the motion estimation techniques in Section 3.3.

Reference	DOF	Method	Track length (m)	Translation error, end position
Nistér et al. (2004)	6-DOF	stereo FW, 3D-2D corners, RANSAC	186	1.1%
Howard (2008)	6-DOF	stereo FW, 3D-2D corners, Max inlier set	400	0.25%
Kitt et al. (2010)	6-DOF	stereo FW, 3D-2D corners, blobs	24	1.3%
Geiger et al. (2011)	6-DOF	RANSAC, Kalman stereo FW, 3D-2D corners, blobs, RANSAC	KITTI	2.4%*
Cvišić & Petrović (2015)	6-DOF	stereo FW, 3D-2D corners, blobs	KITTI	0.88%*
Hansen et al. (2011a)	3-DOF	5-point, RANSAC stereo DW, 3D-2D corners, RANSAC, Model	30	0.85%
Nourani-Vatani & Borges (2011)	3-DOF	mono FW, 2D-2D Area, Model	100	7%
Ericson & Åstrand (2008)	2-DOF	stereo DW, 3D-3D Area, Model	0.6	0.7%
Piyathilaka & Munasinghe (2011)	3-DOF	mono DW, 2D-2D Area, Model, opt flow	35	17%
Jiang et al. (2014)	2-DOF	stereo DW, 3D-3D Corners, ICP	2500	0.2%

* These values were not measured at the end position; instead, they represent averages over several tests and sub-distances according to the method used in the KITTI dataset (Geiger et al., 2012; Geiger, 2016).

selected feature points are used in the motion estimator; they report translation error of 0.88% and rotational error of $0.0022 \text{ deg m}^{-1}$. The KITTI benchmark list reports 2.44% and $0.0114 \text{ deg m}^{-1}$ translation and rotational errors, respectively, for the Libviso method with stereo cameras, as used in this paper.

The Gantry and Libviso methods are tested on both simulated data and on real field data captured by a mobile robot. Figure 1 shows some paths of the real field data. Using both simulated and real field data allows the impact of different design choices to be evaluated, improving our understanding of how various parameters and settings affect the VO results in an agricultural field environment.

3. Visual odometry pipeline

A comprehensive two-part tutorial on VO (Scaramuzza & Fraundorfer, 2011; Fraundorfer & Scara-

muzza, 2012) has presented the steps in the VO pipeline (see Figure 2). The present work follows the structure of that pipeline, treating each step in a separate section.

3.1. Image sequence

The algorithm input is an image sequence from the robot's onboard cameras. This step involves design parameters such as frame rate, resolution, camera type, camera placement, camera calibration, and image preprocessing. Most real-time algorithms work with a resolution of up to 640×480 , while the frame rate is highly dependent on camera placement and maximum speed. Overlap between consecutive frames is required so that points can be correctly matched. The two most common camera placements are facing either forward (Nistér et al., 2004; Howard, 2008; Geiger et al., 2012) or downward (Hansen et al., 2011b,a; Nourani-Vatani & Borges, 2011; Jiang et al., 2014). Besides regular-

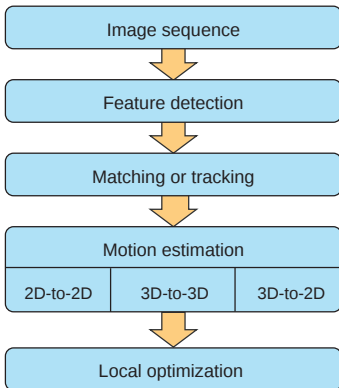


Figure 2: VO pipeline as presented in Scaramuzza & Fraundorfer (2011).

perspective cameras, VO can also use omnidirectional cameras (Corke et al., 2004; Scaramuzza & Siegwart, 2008). The use of stereo cameras enables triangulation, which simplifies the algorithm. VO with monocular cameras can only estimate the position up to a scale factor, so additional assumptions, such as camera height, are required.

3.2. Feature detection and matching

There are two approaches to detection and matching, feature based and area based. The feature-based approach is the most common and is presented as the second step in the VO pipeline. It relies on finding salient points that can be described with unique descriptors that are matched between the images. Assuming that trackable feature points are distributed throughout the image and that these feature points can be correctly matched, this method allows estimation of more degrees of freedom at lower computational cost than do methods using area matching. It also allows more complex transforms and is less sensitive to perspective change. The main drawback is that the method relies completely on the presence of a high number of correctly matched features. A rule of thumb is to adapt the detector so that 1000 features are found in a 640×480 -pixel image (Fraundorfer & Scaramuzza, 2012). A few hundred of these will be matched to each other, but many of these matches will be outliers.

The area-based approach is an appearance-based approach in which the pixel values are matched directly between consecutive images to find correspondence. A template is used, commonly defined as a neighbourhood of fixed points (Nourani-Vatani & Borges, 2011). The matching process searches within a specified area to find the position where the template is most similar to the image. The method is most effective when images differ only by translation, i.e., 2 DOF, but it can also handle rotation and affine transforms at the cost of calculation time (Zitova & Flusser, 2003). In VO, this method is commonly used with downward-facing cameras (Zaman, 2007; Hansen et al., 2011b,a; Nourani-Vatani & Borges, 2011) or with omnidirectional cameras to determine orientation (Scaramuzza & Siegwart, 2008). The main disadvantages of this approach are high computational cost and sensitivity to illumination change. Matching entire images is very time consuming and is usually avoided by selecting one or several smaller regions in the image. That requires a priori information about camera motion and may lead to tracking problems. The advantage of this method is that, compared with the feature-based approach, it can handle images with less texture (Zitova & Flusser, 2003).

3.3. Motion estimation

The next step in the VO pipeline, motion estimation, makes use of points corresponding to each other in consecutive frames. Three options are available in the pipeline, i.e., 2D-2D, 3D-2D, and 3D-3D, depending on how the motion is estimated from the matched points. In the 2D-2D option, commonly used with monocular cameras, the strategy is to estimate the essential matrix and then calculate the 6-DOF motion from that. In the 3D-2D option, commonly used with stereo cameras, 3D points calculated from the first frame using stereo triangulation are moved and then projected back onto the 2D image of the second frame. Minimizing the reprojection error gives the correct motion. Finally, in the 3D-3D method, two 3D point clouds are calculated from first and second images. The motion is estimated from the difference between the two point clouds. The points of interest are those that conform to the rigid-body motion of the camera, i.e., points fitting a model that captures the relative motion between two frames. All other points, such as points on other moving objects or just false matches, are classified as outliers. An efficient method to find valid points in the presence

of outliers is RANSAC (Fischler & Bolles, 1981), which is effective even with a high number of outliers (Fischler & Bolles, 1981; Hartley & Zisserman, 2004). Most previous research in the VO area focuses on finding the best model and optimizing the search for valid points, so that the system can be robust to a high number of outliers and offer real-time performance (Nistér, 2005; Scaramuzza et al.).

The number of points required to estimate a change in position depends on how many degrees of freedom are used in the estimation and on the type of estimation method used. Given a calibrated stereo system, just five matched points are enough to calculate 6 DOF up to a scale factor (Nistér, 2003), i.e., the five-point algorithm. While feature-based approaches must address the problem of reducing the number of points, area-based methods instead must increase the number. Usually additional assumptions are made or the DOF are reduced to three to solve the under-determined system. For example, Nourani-Vatani & Borges (2011) used a robot model and assumed that there would be no sideways slippage of the rear wheels, i.e., a nonholonomic constraint. They also assumed zero mean ground-camera distance when estimating 3-DOF motion from a single-camera setup. They demonstrated that the use of a motion model improved the performance. Regardless of the approach, the importance of using valid points for estimating the position should not be neglected. Howard (2008) highlight this issue by noting that even a single mismatch, i.e., false motion estimate, can introduce more error into the VO estimation than can a thousand correct matches.

3.4. Local optimization

The final step in the VO pipeline is local optimization, which is a technique for minimizing the error over several consecutive frames. The idea is to remove the impact of false motion estimates so that the total error is reduced. Expanding this step converts VO into visual simultaneous localization and mapping (VSLAM), in which a complete map is built and each frame is matched against a global map, i.e., global instead of local optimization. VO applications typically use a few frames for optimization, so the optimization is local. Another option is to optimize when a certain motion or change is detected, allowing the drift to be eliminated when the camera is not moving.

4. Experimental design

This section describes the selected VO algorithms in more detail and highlights the differences between them. The methods are compared according to the steps of the VO pipeline, as shown in Table 2. In the interest of readability, the methods are described only briefly here; for greater detail, the reader is referred to the original publications (Jiang et al., 2014; Geiger et al., 2011).

4.1. Visual odometry theory

Consider a 6-DOF VO system in which the relative pose change of the camera between consecutive frames is modelled as a rigid motion transform. The transform can be written as shown in Equation 1 (Scaramuzza & Fraundorfer, 2011):

$$dTr = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} \quad (1)$$

where $\mathbf{R} \in SO(3)$ is the rotation matrix and $\mathbf{T} \in \mathbb{R}^{3 \times 1}$ is the relative translation. The cumulative pose at frame k can be obtained using Equation 2:

$$C_k = C_{k-1} dTr \quad (2)$$

These are the basic equations of the odometry, and the goal is to find dTr and C_k for each frame. It should be mentioned that the transformation matrix is an overdetermined system and can be rewritten using Euler angles as $\mathbf{v} = (x, y, z, roll, pitch, yaw)$.

4.2. Camera setup

The two evaluated methods both use a feature-based approach. Only stereo perspective cameras are considered, which simplifies the calculation of the distance to the points. Equation 3 shows how the disparities between corresponding points in the left and right images are connected to the distance to the points:

$$Z = f \frac{b}{d} \quad (3)$$

where b is the baseline between cameras and $d = x_l - x_r$ is the stereo disparity between the left and right images.

Furthermore, the images must be corrected for lens distortion and rectified so that the epipolar lines are parallel to the image x -axis, i.e., both intrinsic and extrinsic camera parameters are known.

Table 2: Evaluated algorithms compared according to visual odometry pipeline steps.

Method	Image sequence	Features	Matching	Motion estimation	Local optimization
Gantry	Stereo DW	Harris	Matching and tracking	Iterative closest point Smooth motion filter	None
Libviso	Stereo FW/DW	Blob & corners	Quad- matching	RANSAC 3D-point weight	Pose graph

The standard procedure for calibrating the camera is used, as explained in Section 5.3.

One design parameter is whether a forward- or downward-facing camera should be used. The setting of this parameter is analysed using a theory usually associated with optical flow (Trucco & Verri, 1998). A camera that moves in a static environment creates a motion field representing the camera ego-motion. Let $\mathbf{T} = (T_x, T_y, T_z)$ denote the translational velocity relative to the static 3D environment and $\mathbf{\Omega} = (\omega_x, \omega_y, \omega_z)$ denote the angular velocity of the camera. The relationship between the projected 2D flow field at image point (x, y) and the camera’s 3D motion can be expressed as shown in Equation 4-5:

$$v_x = \underbrace{\frac{T_z x - T_x f}{Z}}_{\text{Translation}} - \underbrace{\omega_y f + \omega_z y + \frac{\omega_x x y}{f} - \frac{\omega_y x^2}{f}}_{\text{Rotation}} \quad (4)$$

$$v_y = \underbrace{\frac{T_z y - T_y f}{Z}}_{\text{Translation}} + \underbrace{\omega_x f - \omega_z x - \frac{\omega_y x y}{f} + \frac{\omega_x y^2}{f}}_{\text{Rotation}} \quad (5)$$

where Z defines the distance to the point and f is the focal length of the camera. Note that these equations can be divided into two parts, the first depending on the camera translation and the second on the camera rotation.

Equation 3 shows that depth Z is inversely proportional to the disparity. The error of the image point position can be modelled as Gaussian (Matthies & Shafer, 1987), so the Z error will be large for points far from the camera. At the same time, Equations 4 and 5 show that points far from the camera, i.e., $Z \rightarrow \infty$, do not generate any flow in the translational part of the flow field, since the first term can be neglected. Large flow fields for distant points are therefore likely to result from rotation of the camera. In contrast, points near the camera will contain both translation and rotation.

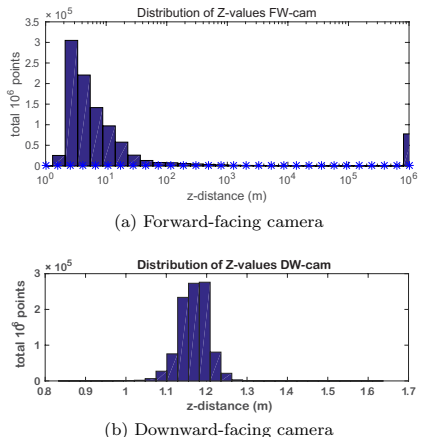


Figure 3: Distribution of distance to points: histogram of 10^6 points; maximum distance is limited to 10^6 m.

These points require better depth accuracy in order to provide good ego-motion estimation.

Consider a vehicle driving forward across an open field. If the cameras are mounted perpendicular to the ground, all points will be a similar distance from the camera. All points can be considered near the cameras and the error of the 3D points will be fairly small. The 3D points will be much wider ranging if the cameras are mounted facing forward, including both points near the vehicle and distant points at the horizon. Figure 3 shows the depth distribution of one million matched 3D points from one of the real field tests. The images are similar to the two sample images shown in Figure 6. The best theoretical solution for selecting the camera angle is not obvious, because it depends greatly on the error of the 3D points. This error in turn depends on the field structure and the algorithm’s ability to detect robust points.

This implies that there might be an advantage to estimating the translational part separately from

the rotational part, which the Gantry method does. Weights are instead used to control the impact of distant points on the motion estimation. Libviso uses weights in RANSAC to reduce the impact of distant points, while Gantry selects only points within a certain distance.

One requirement of VO is that consecutive images overlap so that feature points in one image can be found in the next. This is usually not a problem with forward-facing cameras because the focus of expansion is usually visible in the image. For a downward-facing camera, however, this is highly dependent on the maximum velocity. Equation 6 shows the relationship between the maximum velocity and overlap:

$$overlap = 1 - \frac{v_{max}}{2hf_s \tan(\frac{\alpha}{2})} \quad (6)$$

where v_{max} is maximum velocity ($m \cdot s^{-1}$), h is the camera height above the ground (m), f_s is the frame rate, and α is the field of view of the camera.

4.3. Feature detection

Harris corners are used in the Gantry method (Harris & Stephens, 1988), where they are matched to each other using the correlation of an 11×11 -pixel neighbourhood. Points near each other are suppressed if the corner response is not the maximum in a 20×20 -pixel neighbourhood. That limits the maximum number of feature points on a 640×480 -pixel image to 3072. Libviso, on the other hand, detects both blobs and corners. The images are filtered with 5×5 -pixel templates and both minimum and maximum responses are selected as feature points. Finally, Libviso builds a descriptor at two scale levels based on a Sobel-filtered image. This method generates a large number of matches, i.e., up to approximately 5000 for a well-textured 640×480 -pixel image. This number is then reduced in ensuing matching and outlier removal steps, typically to a few hundred.

4.4. Feature matching/tracking

The Gantry approach uses both matching and tracking. First, features are matched as described previously to find stereo correspondence, which gives a 3D point cloud according to Equation 3. The points in the next frame are then tracked using the Lucas Kanade feature tracker (Bouguet, 2000), which calculates a new point cloud.

In the Libviso method, the descriptors are matched using a circular check for correspondence between all four images in two consecutive image pairs. Outliers are removed in two steps: first, by Delaunay triangulation, which removes points near each other; second, by bucketing, which is a technique for keeping only a certain number of points in each area, resulting in a more even distribution of points over the image (Kitt et al., 2010).

4.5. Motion estimation

Even correctly matched points might still belong to other moving objects and not represent camera motion; such points will also be treated as outliers from now on. Both evaluated methods differ in how the motion is estimated and in how these new outliers are detected and removed. The Gantry method estimates motion by linking 3D point clouds to each other using the iterative closest point (ICP) algorithm, i.e., a 3D-3D method according to the VO pipeline. ICP can provide a deterministic solution to the motion estimation problem, instead of the stochastic solution that RANSAC provides. On the other hand, RANSAC can handle more outliers than the ICP algorithm can. The Gantry method tries to solve this problem by removing some of the outliers by adding motion constraints to the points. A smooth motion filter removes all points not conforming to the expected motion of the robot. A weighting for the 3D points describing how much each point pair diverges from the expected motion is used to suppress the impact of unexpected motion. Evaluations of various ICP algorithms are presented by Pomerleau et al. (2013). The present work uses the implementation presented by Geiger et al. (2012), which is based on Arun et al. (1987).

Libviso uses a more standard RANSAC approach that minimizes the reprojection error of 3D points, i.e., a 3D-2D method according to the VO pipeline. RANSAC is an iterative method in which three 3D points are randomly selected from the previous image pair. The relative motion, dTr , is solved for using the Levenberg-Marquardt algorithm (Marquardt, 1963) so that the projection error is minimal. The motion is then applied to all points to determine the number of inliers, i.e., points with reprojection error below a threshold.

4.6. Local optimization

The Libviso method includes no local optimization, which the user must determine. The Gantry

method uses pose graph optimization (Fraundorfer & Scaramuzza, 2012) with five frames. The motions between multiple frames are calculated according to a scheme and the size of the error determines whether estimation over several frames or cumulative estimation of consecutive frames should be used. Using five frames for optimization requires three more pose calculations, which reduces the system’s real-time performance.

5. Experiments

The experiments were evaluated on two types of datasets, one comprising synthetic images generated by a simulator and one comprising real field data. The advantages of using the simulated data are that they enable full control of the robot path and provide very accurate ground truth. In that way, certain parameters can be evaluated by creating special datasets, such as driving straight forward on a plane surface with different camera baselines or camera heading angles. The real field datasets come both from the KITTI dataset (Geiger et al., 2012), which represents the urban/road scenario, and from our own experiments representing the agricultural scenario. A mobile robot based on an electrical wheelchair was used to acquire data on an agricultural field (see Figure 4). The robot has a total of four cameras mounted in two stereo pairs, one facing forward and the other facing downward; there is also an omnidirectional camera at the top that was not used in these experiments. Next to the omnidirectional camera are a real-time kinematic GPS (RTK-GPS) and an inertial measurement unit (IMU) used for determining ground truth. Additional calibration runs were conducted to calibrate the cameras’ positions on the robot.

5.1. Simulator

The simulator was developed using the Unity game engine (Unity Technologies, Bellevue, WA, USA) in which suitable 3D environments can be built. Cameras are positioned as in the actual robot, with one stereo pair facing forward and one facing downward. The environment is designed to be representative of a corresponding real environment, including mountains, trees, and a blue sky with some clouds (see Figure 5). Cameras are mounted at a height of 1 m from the ground.

The path of the camera can be set arbitrarily, but it is particularly interesting to follow exactly

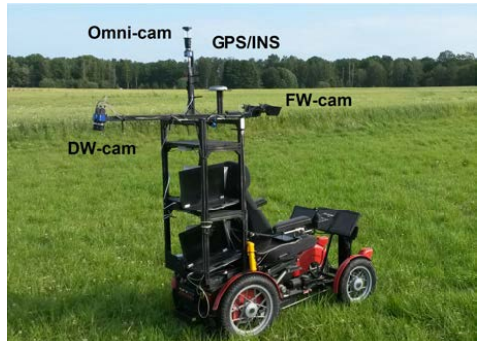


Figure 4: Experimental platform with sensor placement: FW-cam is the forward-facing stereo camera and DW-cam is the downward-facing stereo camera. The omnidirectional camera was not used in the research reported here.

the same path as in the real experiments. In that case, the simulator is fed with the ground truth from the real experiment. The scene is based on the same path so the environment is similar to the real world. It is also possible to drive in a perfectly straight line on a perfect plane to evaluate various parameters such as sideways slippage, pitch, and roll. This could also be done by others to benchmark their setups. The simulator output is a sequence of synthetic images. Figure 5 shows a sample image from one of the KITTI datasets and a corresponding scene in the simulator, the difference being that no houses or cars are included in the simulation. These are replaced with trees, which provide a similar distribution of tracking points in the image.

5.2. Mobile robot

The sensor placement on the robot can be seen in Figure 4. The cameras used are mvBlueFox-120a (Matrix Vision, Oppenweiler, Germany) configured to capture 640×480 -pixel grey-scale images at 20 Hz. The stereo baseline is 200 mm for the forward-facing and 50 mm for the downward-facing cameras. The forward-facing cameras are tilted 12 degrees towards the ground to minimize the amount of sky in the images and to increase the amount of trackable ground. The downward-facing cameras are tilted slightly backwards (98 degrees) to exclude the robot from the cameras’ field of view. Sun covers are used to minimize lens flare, which appears when driving towards the sun. The downward-facing cameras are



(a) Sample image from the KITTI dataset.



(b) Corresponding simulated image.

Figure 5: Sample image from one of the forward-facing cameras in the simulated environment.

mounted approximately 1 m behind the forward-facing cameras at a height of 1.1 m. Only a small part of the robot is visible at the top of the image, but shadows from the robot enter the ground cameras' field of view when driving towards the sun. This will reduce the number of feature points found, as the contrast in the shaded region may be too low.

The RTK-GPS is mounted slightly behind the forward-facing cameras on the top of the robot, together with the IMU. The RTK-GPS is based on a LEA-4T GPS receiver (u-blox, Thalwil, Switzerland) and a high-performance RTK antenna. The system consists of a base station and a rover unit, both using libRTK (Takasu & Yasuda, 2009; Takasu, 2013). These are connected using mobile Internet and the base station is placed within 1 km of the field. A 9-DOF Razor IMU (sparkfun, Niwot, CO, USA) is used to give the orientation of the robot.

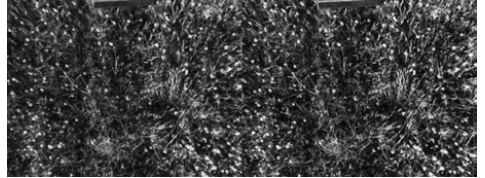
All sensors are hardware synchronized. The RTK-GPS runs at 10 Hz while all other sensors are sampled at 20 Hz. Data are collected from all on-board sensors and stored on disks for offline analysis. An additional GPS unit is attached to provide time measurements of the synchronization pulses, providing valid time stamps for all measurements.

5.3. Calibration

The calibration procedure consists of calibrating both the cameras and the sensor positions on the robot. The cameras are calibrated using a standard



(a) Sample images from the forward-facing stereo camera.



(b) Sample images from the downward-facing stereo camera.

Figure 6: Sample images from stereo cameras: left- and right-hand images are placed side by side; resolution is 640×480 pixels.

procedure and the Matlab toolbox (Bouquet, 2008). The forward-facing cameras use an 8×6 chessboard in which each square is 100×100 mm. The distance between the cameras and the chessboard was altered between 1 and 5 m. The calibration data are used for correcting radial and tangential distortion in the captured images. The images are also rectified so that the epipolar lines are parallel to the camera's x -axis. The calibration of the simulated data is calculated using the camera settings in the simulator. The results are verified by adding a 100×100 -mm chessboard to the simulator and then following the same procedure as in the real case.

The sensor positions are determined in two steps. First, the sensor positions and rotations on the robot are measured approximately using a scale on the robot. Second, the robot drives in a calibration pattern (comprising straight, turning, and figure-of-eight paths) and positions are estimated using VO. The relative movement of each sensor is compared with the relative movement of the robot and nonlinear optimization is performed using the Levenberg-Marquardt algorithm. The manually measured values are used as initial values for the optimization.

5.4. Simulated datasets

The first datasets are created to evaluate different design options and to highlight differences between

the agricultural and urban/road scenes. The design options and environmental conditions evaluated are as follows:

- Baseline – affects the accuracy of the 3D points. The KITTI dataset uses a 0.53-m baseline on its forward-facing cameras, while the dataset used in evaluating the Gantry method uses a 0.05-m baseline on its downward-facing cameras. The baselines used in the experiments reported here are 0.20 m for the forward-facing cameras and 0.05 m for the downward-facing cameras.
- Image resolution – affects the number of feature points that can be extracted and their distribution in space. The KITTI dataset uses images that have a higher resolution of 1241×376 pixels as compared with the 640×480 -pixel resolution used in the Gantry dataset. The cameras used in the present research have a resolution of 640×480 pixels.
- Feature point distribution – affects motion estimation accuracy. The KITTI dataset comes mainly from an urban environment with feature points distributed throughout the image and the focus of expansion near the image centre. The agricultural scene represents an open field with few or no feature points in the sky portion of the image and a focus of expansion located off centre. The Gantry data use downward-facing cameras, which are assumed to give images with many feature points distributed evenly throughout the image.
- Camera angle – affects the number of feature points and the maximum velocity. The camera facing directly forward has an angle of 0 degrees and the camera facing downward has one of 90 degrees.
- Frame rate – affects the maximum velocity. A higher frame rate enables higher velocity, but more frames may lead to more cumulative error per metre. Both the KITTI and Gantry datasets are sampled at 10 Hz, while a frame rate of 20 Hz is used in our experiments.
- Algorithm selection – comparison of the feature extractor and motion estimator in the Gantry and Libviso methods.

The feature point distribution is an environmental parameter that describes the properties of the

feature points in a scene. It is included for evaluating the impact of changing from the urban/road to an open field environment. Trees near the vehicle give a different point distribution from that of an open agricultural field.

The simulator scene is created to represent the KITTI 00 sequence data, (Geiger, 2016), the path being determined from the ground truth of this dataset. The first parameter setup corresponds to the settings of the vehicle used when collecting the data. This means that the results of the real and simulated data should correspond to each other. In the next step, one design option is changed for each dataset so that the impact of doing so can be evaluated.

The next simulated dataset is when the cameras move back and forth across a field, driving 40 m straight ahead and then making a headland turn. The velocity is set to 50 mm frame^{-1} when driving straight and 30 mm frame^{-1} when turning. These velocities correspond to those used in real field tests. The track length is 544 m and 13,000 frames are acquired; in total, 10 headland turns of 180° were made.

5.5. Real field dataset

The field tests were performed on two different fields, one fodder field and one oat field. The fodder field consisted mainly of clover, grass, and dandelions and the average plant height was estimated to be 200 mm. The oat plants in the oat field were at various growth stages, so they ranged in height from 20 to 500 mm. One part of the field was waterlogged during regular sowing, so this part had recently been resown. The mobile robot was controlled manually during the experiment, traversing the field in a regular back-and-forth pattern. Figure 1 shows the test fields, the left-hand field being the oat field and the right-hand field the fodder field. The figure also shows the surroundings and example tracks from two of the experiments taken from the GPS receiver. Each field was approximately $300 \times 200 \text{ m}$. The experiments are summarized in Table 3; the data were analysed offline. Figure 6 shows sample images from both the forward-facing and downward-facing cameras.

The test on the oat field was conducted during light rain in daytime. Despite using covers to protect the lenses, one raindrop hit the left lens of the forward-facing camera, staying there for half the test. In addition to the raindrop, other real objects appeared, such as a car running along the

Table 3: Datasets from the agricultural field.

Test label	Field type	Weather	Distance	Frames	Max. speed (mm frame ⁻¹)	Max. rotation (deg frame ⁻¹)
Field 1	Oat	rainy	2093	32700	109	0.158
Field 2	Fodder	sunny	1838	33400	84	0.196
Field 3	Fodder	partly cloudy	2366	58380	66	0.151
Field 4	Fodder	partly cloudy	3065	63000	78	0.144

nearby road and birds taking off when the robot approached. One experiment on the fodder field was performed during the evening when the sun was declining towards the horizon. The low sun was visible to the forward-facing camera, creating lens flares. The sun also created long shadows behind the robot, which are visible in the images from the downward-facing camera. These conditions were not ideal for VO, but are highly representative of the real agricultural case. The other two experiments on the fodder field were conducted during daytime on a partly cloudy day in October. During these experiments, shadows were visible to the forward-facing camera during the headland turns, but no shadows were visible to the downward-facing camera.

5.6. Error calculations

The error is calculated in the same way as in the KITTI evaluation. The entire track is divided into sub-sequences 100, 200, ..., 800 m long. Each sub-sequence is analysed by calculating an error vector between the VO result and the ground truth at the endpoint of the sub-sequence. The translation error is the length of the error vector and is presented in per cent of travelled distance. This measure therefore includes errors in the x , y , and z directions. The rotational error includes errors in terms of roll, pitch, and yaw and is calculated as the angle between the VO heading and the ground truth heading. This error is also divided by the travelled distance, giving a measure in deg m⁻¹.

The total error is calculated by averaging all possible sub-sequences that can be extracted from the track. This gives a more representative value of the error but also requires more accurate ground truth. Another common approach is just to evaluate the measure at the end point. This gives a more stochastic value as it depends on the travelled path. It can also suffer from error cancellation, giving inaccurate results. A detector of wrong motion estimates is included in addition to the error measure.

It determines whether the new estimated position lies within 1 m of the ground truth. If not, it is considered a major error in the motion estimation and is counted. The total number of wrong motion estimates is presented in the results.

6. Results and discussion

This section is structured as follows. Section 6.1 presents VO results for different parameter settings along with related discussion. Both real and simulated data from the KITTI dataset are used in this evaluation. Section 6.2 presents the VO results using both the Gantry and Libviso methods for a simulated agricultural field. Finally, the results of the real field tests are presented in Section 6.3

6.1. Urban environment simulations

The impacts of the design parameters are evaluated on the simulated dataset, with the path and environment corresponding those of the real experiment from the KITTI dataset. To verify the Libviso algorithm, it is first tested on all sequences from the dataset that are provided with ground truth. The results of these 11 sequences give errors of 2.74% and 0.0135 deg m⁻¹, which are slightly higher values than on the KITTI benchmark list, where the results of 21 sequences are reported to give errors of 2.44% and 0.0114 deg m⁻¹. These measures are used in validating the algorithm. Figure 7 shows the 3D path of the simulated KITTI dataset plotted in 2D, and Table 4 shows the results for the simulated dataset with different design options. Between each simulation, only one parameter is changed compared with the reference, except in the combined test, in which two parameters are intentionally changed.

The results of using the simulator for the same track as the real KITTI 00 sequence data primarily indicate that the simulator provides better results than do the real data. This is as expected and validates the use of the simulator. Note that the

Table 4: Visual odometry results of using the Libviso algorithm in simulation using the KITTI 00 sequence data for a total of 4540 frames. The error is measured using the KITTI method. Wrong motion estimates are the numbers of frames with translation error exceeding 1 m between two consecutive frames.

Test setup	Translation error	Rotation error (deg m ⁻¹)	Matches (average)	Inliers (average)	Wrong motion est.
Reference real KITTI 00 1241 × 376, baseline = 0.53, Cam = 0°	2.96%	0.0146	211	88%	0
Reference simulated KITTI 00 1280 × 480, baseline = 0.53, Cam = 0°, frames = ×1	0.76%	0.0029	420	87%	0
Baseline = 0.05 m	10.80%	0.0493	442	46%	53
Baseline = 0.2 m	1.77%	0.0040	444	78%	20
Baseline = 1 m	0.65%	0.0027	415	89%	0
Images at 640 × 480	1.55%	0.0035	226	92%	19
Less feature points 1	3.23%	0.0028	430	90%	56
Less feature points 2	5.50%	0.0030	423	91%	125
Cam 12°	0.76%	0.0030	435	88%	0
Cam 20°	2.32%	0.0022	436	88%	40
Cam 30°	4.63%	0.0025	433	90%	107
Cam 45°	12.3%	0.0052	437	94%	367
More frames ×1 (4540)	0.75%	0.0030	435	88%	0
More frames ×2 (9080)	3.99%	0.0167	465	92%	5
More frames ×5 (22700)	4.41%	0.0186	476	93%	1
More frames ×10 (45400)	9.53%	0.0508	479	94%	7
Combined baseline = 0.2, 640 × 480	2.42%	0.0048	230	88%	30

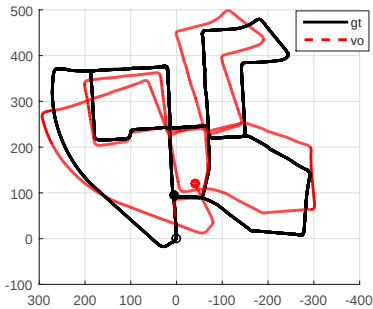


Figure 7: Resulting 3D plot from using the Libviso algorithm on simulated KITTI 00 sequence data; black solid line = ground truth, red dashed line = visual odometry results, open circle = start position, solid circle = end position.

number of matched features increases by a factor of two, mainly because fewer points are rejected as outliers. There are no wrong motion estimates in this test.

A wider baseline gives better VO results due to the increased accuracy of the 3D points. There are fewer inliers for short baselines and more motion estimation errors, indicating that the positions of the 3D points are less accurate. That means that the wider baseline is preferred as long as the points can be matched without leading to perspective error. In this case they can be matched, which is seen by the high number of matches found.

The KITTI dataset uses wide images of 1241 × 376 pixels. This test also indicates that the VO results display increased error when the image size is decreased to 640 × 480 pixels. The number of feature points decreases by half due to the bucketing function of the Libviso method. As several parameters are not changed at the same time in these experiments, the same bucket size and same number of points in each bucket are used. This means

that when the image size is decreased by a factor of two, so is the number of features, leading to less accurate results.

The attempt to decrease the number of features actually fails, as the average number of features is still the same, see Table 4. However, the overall VO results are less accurate due to more false motion estimates, indicating that the extracted features are less robust in these scenes. The high average number of inliers also confirms that some failed frames contributed to this error.

The impact of different camera angles indicates that the error is similar at 0° and 12° , but that there is increased error at higher angles, i.e., when the camera is pointing more towards the ground. At higher angles the maximum velocity is violated, resulting in little or no overlap between the images and leading to incorrect motion estimates for some frames. There are no wrong estimates at 0° and 12° , but at 20° and above the number of wrong estimates increases with the angle. Note that the rotational error remains more or less the same whatever the angle.

One way of handling small overlaps is to increase the frame rate. This gives more images for the same path but increases the cumulative error as well. The simulation results indicate that increasing the frame rate increases the number of both features and inliers, though the translation and rotational errors still increase. In particular, the rotational error increases from 0.003 to 0.05 deg m^{-1} as the number of frames increases by a factor of 10, greatly affecting the final VO results.

Finally, there are two combined tests in which the baseline and image size are changed at the same time. As expected, the error is higher than when changes are made individually, though it is lower than the sum of the errors resulting from the individual changes.

The urban environment simulations indicate that the errors belong to three categories, i.e., *accuracy errors*, *motion estimation errors*, and *cumulative errors*, each of which must be considered when designing a VO system. Accuracy errors are those errors resulting mainly from errors in calibration and from noise in the estimated 3D points. The noise level in these experiments can be seen by comparing all the results with no wrong motion estimates. The best results are obtained when the camera is pointing straight ahead, which is due to lower rotational error. This highlights the importance of properly estimating the heading. Motion estimation errors

occur when any of the VO limits are violated, such as the maximum velocity and minimum number of feature points representing camera motion. There is a strong correlation between the results of the above experiments and the number of motion estimation errors in all experiments except the test of an increasing number of frames. This leads to the last category, cumulative errors. This type of error depends greatly on the number of frames, but is naturally connected to the accuracy error. If there were no accuracy errors, there would be no cumulative errors. This type of error affects both the translational and rotational part of the error and increases with the number of frames used.

6.2. Agricultural field simulations

These results are from the simulated dataset representing the agricultural scenario. The path is set to a common back-and-forth pattern on a perfect plane. The camera angle is set to 12° for the forward-facing cameras and to 90° and 94° for the downward-facing cameras. This dataset is evaluated using both the Gantry and Libviso algorithms. The multiframe optimization used by the Gantry algorithm is turned off in this test so the results of the two methods can be compared. It has already been demonstrated that multiframe optimization can improve performance (Jiang et al., 2014). The results are shown in Table 5 and Figure 8.

The VO simulations of the agricultural field produce the expected results when using the forward-facing camera and the Libviso method. These results are consistent with previous results of KITTI simulations in which there were fewer features and more frames (i.e., 13,500 instead of 4500). On the other hand, the path followed is simpler with no roll, pitch, or height changes. The results with the downward-facing camera are very similar to those with the forward-facing camera when the Libviso method is used.

One test run stands out for its extremely low error. This is the run when the Gantry method is used with downward-facing cameras positioned exactly perpendicular to the ground, which is a perfect plane. The results indicate that all points in the point cloud are estimated to be the same distance from the camera, which leads to correct measurement of the roll, pitch, and height. There are also no outliers in the point cloud, so the translation and yaw can be correctly estimated. On the other hand, if the camera is tilted slightly, i.e., to 94° , height estimation errors will occur, which will be

Table 5: Visual odometry results for a simulated agricultural field; the Gantry method does not use multiframe optimization in these tests, and there are no motion estimation errors in these tests.

Test setup	Translation error	Rotation error (deg m ⁻¹)	Matches (average)	Inliers (average)	Error end pos.
FW 12° Libviso	3.29%	0.0612	234	99.2%	5.07%
DW 90° Libviso	3.26%	0.0588	245	99.8%	5.84%
DW 94° Libviso	3.65%	0.0632	238	99.7%	4.91%
DW 90° Gantry	0.04%	0.0008	431	100%	0.02%
DW 94° Gantry	5.14%	0.1248	433	100%	8.73%

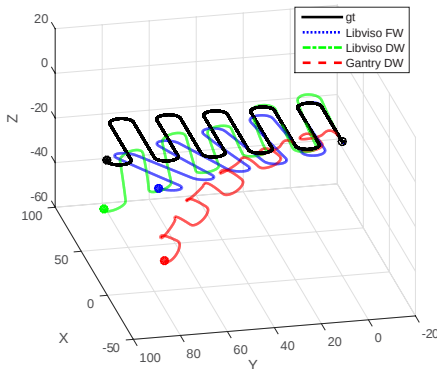


Figure 8: 3D plot resulting from using the Libviso and Gantry algorithms on a simulated agricultural field; camera angle is 90°, open circle = start position, solid circle = end position.

slightly worse than if the Libviso method is used. When the libviso method is used there are no major differences between downward-facing and forward-facing cameras based on these results.

The same data are also used for comparing different combinations of the feature extractors and motion estimators used in the Libviso and Gantry methods. The results are shown in Table 6. It can be seen that the Gantry feature extractor (i.e., using Harris corners) provides more accurate features than does the Libviso method, which uses several types of corners and blobs. This is as expected, as Harris corners are known to be accurate (Fraundorfer & Scaramuzza, 2012). The bucketing in the Libviso method decreases the total number of matches, which also decreases the accuracy. Gantry’s ICP method for estimating motion is more sensitive to outliers than is the RANSAC method used in Libviso. This makes the results of the Libviso feature detector and the Gantry motion estimator (ICP) worse than the others.

To summarize, the experiments using simulated agricultural field data indicate that the method that performs outstandingly under ideal conditions produces slightly worse results in the presence of outliers or a more realistic environment.

6.3. Agricultural field experiments

The results of the real field experiments are shown in Table 7. In total, four test runs were conducted, each several km long. There are high numbers of frames in these tests, which makes the cumulative error dominant. To provide a more reasonable number of frames and to reduce the cumulative error, an additional evaluation of these datasets was conducted using a sub-sample of the data set. In this evaluation, every 10th frame was analysed for all tests except the Field 2 test, in which every 5th frame was analysed. The results are shown in Table 8. A plot of the Field 2 test run is shown in

Table 6: Comparison of the feature extractors and motion estimators: visual odometry results for a simulated agricultural field; the angle of the downward-facing cameras is 90° .

Test setup		Translation error	Rotation error (deg m ⁻¹)	Matches (average)	Inliers (average)
Feature	Motion				
Gantry	Gantry	0.04%	0.0008	431	100%
Gantry	Libviso	0.23%	0.0063	431	100%
Libviso	Gantry	4.11%	0.1150	245	99.6%
Libviso	Libviso	3.26%	0.0588	245	99.8%

Figure 9, showing both the original sampling and the sub-sampling results.

The main error source is poor estimation of pitch and roll. In combination with long accumulation, as in the real field experiments, these errors grow until the results are unusable. Still, it can be seen that the forward-facing camera performs much better than does the downward-facing camera, particularly in terms of rotation. The results with the downward-facing cameras are similar between the methods. The pitch and roll are very difficult to estimate and the algorithms have difficulties separating pitch from forward motion when driving forward. This is particularly true with the Gantry method, in which the points are tracked and all points used in the estimation are gathered along the bottom edge of the image. The pitch is then estimated using 3D points with fairly large height errors and a narrow distribution in the motion direction. This leads to poor motion estimation with a pitch bias, causing a huge cumulative VO error. Nourani-Vatani & Borges (2011) stated that pitch and roll cannot be estimated using VO with a camera positioned perpendicular to a flat plane; their solution was to incorporate other sensors for full 6-DOF estimation.

The results can easily be improved by selecting a sub-sample of the data-set, i.e., using a lower frame rate (see Table 8). This can be done as long as the minimum overlap is not violated, and for these experiments it can only be done with the forward-facing cameras. The results improve significantly and the best results correspond to the KITTI results (see Table 4), taking into account that the images have different resolutions, different baselines, etc. (see Fields 2 and 3 in Table 8). These results also include some motion outliers, i.e., wrong motion estimates. The worst case is found in the Field 3 experiment, in which the roll is estimated to be 16.3° instead of 1.3° . That single error increases the total translation error from 3.32% to 3.91%, veri-

fying Howard’s (2008) claim as to the great impact of even a single motion estimation error. A motion model may reduce such errors and improve the performance.

The Field 1 experiment used a small number of features, only 70 on average. The rainy weather resulted in a very challenging dataset with low-contrast images, leading to larger error in the motion estimation.

7. Conclusions

This paper aims to improve our knowledge of the error resulting when visual odometry is applied in an agricultural field environment. In contrast to the urban scene, the agricultural scene is more open and provides a different distribution of feature points. The findings indicate that high-resolution images captured from forward-facing stereo cameras mounted with a large baseline would give the best results with the least error. The main error source is the accumulation of many small errors in the motion estimator. The algorithm must be able to reduce this type of error accumulation either by adapting the frame rate to minimize the number of frames used, or by incorporating other sensors that can compensate for the drift over time. The algorithm must also detect and handle large errors in the motion estimation to prevent motion outliers. The findings also illustrate the difficulties of estimating roll and pitch with a downward-facing camera, as it produces feature points with a disadvantageous distribution. The forward-facing camera setup offers better heading estimation, which has a great positive impact on the VO results.

The Gantry method was demonstrated to be very accurate under optimal conditions, but it is sensitive to outliers and is less robust than the Libviso method, which is less accurate. Its consistent results make the Libviso method more predictable, which is an advantage over the Gantry

Table 7: Results of field test on a fodder field.

Test setup	Distance (m)	Number of frames	Translation error	Rotation error (deg m ⁻¹)	Matches (average)	Inliers (average)
FW Libviso						
Field 1, oat field	2093	32700	23.5%	0.1858	139	87%
Field 2, fodder field	1838	33400	8.69%	0.1579	199	95%
Field 3, fodder field	2366	58380	16.3%	0.2293	221	92%
Field 4, fodder field	3065	63000	37.3%	0.3892	206	85%
DW Libviso						
Field 1, oat field	2093	32700	49.9%	0.4304	191	55%
Field 2, fodder field	1838	33400	22.8%	0.3725	217	89%
Field 3, fodder field	2366	58380	25.3%	0.3765	222	89%
Field 4, fodder field	3065	63000	37.1%	0.4031	222	89%
DW Gantry						
Field 1, oat field	2093	32700	46.8%	0.4535	397	51%
Field 2, fodder field	1838	33400	21.0%	0.3716	368	70%
Field 3, fodder field	2366	58380	29.7%	0.4755	403	87%
Field 4, fodder field	3065	63000	46.4%	0.4356	412	80%

Table 8: Results of field test on a fodder field with a sub-sample of the data set.

Test setup	Distance (m)	Number of frames	Translation error	Rotation error (deg m ⁻¹)	Matches (average)	Inliers (average)
FW Libviso						
Field 1, oat field, rainy	2093	3270	17.1%	0.0404	70	79%
Field 2, fodder field, sunny	1838	6680	3.76%	0.0482	184	90%
Field 3, fodder field, cloudy	2366	5838	3.91%	0.0365	174	83%
Field 4, fodder field, cloudy	3065	6300	10.8%	0.0898	152	66%

method. The real experiments produced poor results in runs in which the total number of frames exceeded 30,000. The best full 6-DOF position estimation results were obtained with a 1.8-km run using a sub-sample of 6680 frames; in this run, the translation error was 3.76% and the rotational error 0.0482 deg m⁻¹.

8. Acknowledgements

The authors would like to thank Mariestad Municipality for providing access to the agricultural test fields, and Anna Syberfeldt for her constructive comments and suggestions on this work.

References

- Arun, K. S., Huang, T. S., & Blostein, S. D. (1987). Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9, 698–700. URL: <http://dx.doi.org/10.1109/TPAMI.1987.4767965>. doi:10.1109/TPAMI.1987.4767965.
- Bouguet, J.-Y. (2000). Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, .
- Bouguet, J. Y. (2008). Camera calibration toolbox for Matlab. URL: http://www.vision.caltech.edu/bouguetj/calib_doc/.
- Corke, P., Strelow, D., & Singh, S. (2004). Omnidirectional visual odometry for a planetary rover. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* (pp. 4007 – 4012 vol.4). volume 4. doi:10.1109/IROS.2004.1390041.
- Cvšić, I., & Petrović, I. (2015). Stereo odometry based on careful feature selection and tracking. In *European Conference on Mobile Robots 2015*. doi:10.1109/ECMR.2015.7324219.
- Ericson, S., & Åstrand, B. (2008). Visual odometry system for agricultural field robots. In *World Congress on Engineering and Computer Science 2008, International Conference on Intelligent Automation and Robotics 2008* (pp. 619–624). San Francisco, USA. doi:<http://dx.doi.org/10.1109/WCECS.2008.26>.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24, 381–395. URL: <http://doi.acm.org/10.1145/358669.358692>. doi:<http://doi.acm.org/10.1145/358669.358692>.

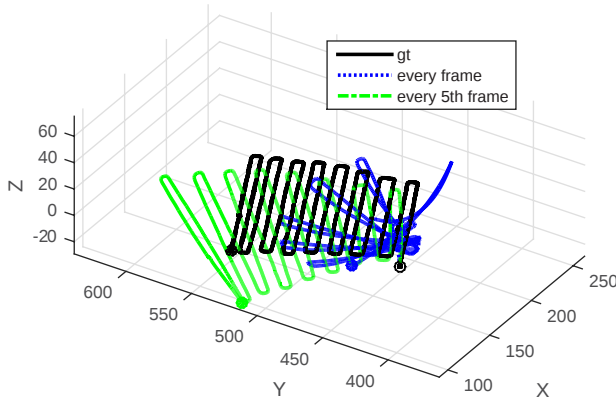


Figure 9: Resulting 3D plot from using the Libviso algorithm with forward-facing cameras on Field 2, showing impact of different frame rates; open circle = start position, solid circle = end position.

- 358669.358692.
- Fraundorfer, F., & Scaramuzza, D. (2012). Visual odometry : Part ii: Matching, robustness, optimization, and applications. *Robotics Automation Magazine, IEEE*, 19, 78–90. doi:10.1109/MRA.2012.2182810.
- Geiger, A. (2015). Libviso2: C++ library for visual odometry 2. URL: <http://www.cvlibs.net/software/libviso/>.
- Geiger, A. (2016). The kitti vision benchmark suite. online. URL: http://www.cvlibs.net/datasets/kitti/eval_odometry.php.
- Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/CVPR.2012.6248074.
- Geiger, A., Ziegler, J., & Stiller, C. (2011). Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*. Baden-Baden, Germany. doi:10.1109/IVS.2011.5940405.
- Hansen, P., Alismail, H., Browning, B., & Rander, P. (2011a). Stereo visual odometry for pipe mapping. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (pp. 4020–4025). doi:10.1109/IROS.2011.6094911.
- Hansen, P., Alismail, H., Rander, P., & Browning, B. (2011b). Monocular visual odometry for robot localization in lng pipes. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (pp. 3111–3116). doi:10.1109/ICRA.2011.5979681.
- Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference* (pp. 147–151). doi:10.1.1.231.1604.
- Hartley, R. I., & Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. (2nd ed.). Cambridge University Press, ISBN: 0521540518.
- Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* (pp. 3946–3952). doi:10.1109/IROS.2008.4651147.
- Jiang, D., Yang, L., Li, D., Gao, F., Tian, L., & Li, L. (2014). Development of a 3d ego-motion estimation system for an autonomous agricultural vehicle. *Biosystems Engineering*, 121, 150–159. URL: <http://www.sciencedirect.com/science/article/pii/S1537511014000373>. doi:<http://dx.doi.org/10.1016/j.biosystemseng.2014.02.016>.
- Kameras als Augen des Jät-Roboters (2015). Kameras als augen des jät-roboters. Markt & Technik.
- Kitt, B., Geiger, A., & Lategahn, H. (2010). Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *IEEE Intelligent Vehicles Symposium*. San Diego, USA. doi:10.1109/IVS.2010.5548123.
- Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11, 431–441. doi:10.1137/0111030.
- Matthies, L., & Shafer, S. (1987). Error modeling in stereo navigation. *Robotics and Automation, IEEE Journal of*, 3, 239–248. doi:10.1109/JRA.1987.1087097.
- Nistér, D. (2003). An efficient solution to the five-point relative pose problem. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (pp. II–195). IEEE volume 2. doi:10.1.1.86.8769.
- Nistér, D. (2005). Preemptive ransac for live structure and motion estimation. *Mach. Vision Appl.*, 16, 321–329. URL: <http://portal.acm.org/citation.cfm?id=1107685.1107693>. doi:10.1007/s00138-005-0006-y.
- Nistér, D., Naroditsky, O., & Bergen, J. (2004). Vi-

sual odometry. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1, 652–659. doi:<http://doi.ieeecomputersociety.org/10.1109/CVPR.2004.265>.

Nourani-Vatani, N., & Borges, P. V. K. (2011). Correlation-based visual odometry for ground vehicles. *Journal of Field Robotics*, 28, 742–768. URL: <http://dx.doi.org/10.1002/rob.20407>. doi:10.1002/rob.20407.

Piyathilaka, L., & Munasinghe, R. (2011). Vision-only outdoor localization of two-wheel tractor for autonomous operation in agricultural fields. In *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on* (pp. 358–363). doi:10.1109/ICIINFS.2011.6038094.

Pomerleau, F., Colas, F., Siegwart, R., & Magnenat, S. (2013). Comparing icp variants on real-world data sets. *Autonomous Robots*, 34, 133–148. URL: <http://dx.doi.org/10.1007/s10514-013-9327-2>. doi:10.1007/s10514-013-9327-2.

Scaramuzza, D., & Fraundorfer, F. (2011). Visual odometry [tutorial]. *Robotics Automation Magazine, IEEE*, 18, 80–92. doi:10.1109/MRA.2011.943233.

Scaramuzza, D., Fraundorfer, F., & Siegwart, R. (). Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *EEE International Conference on Robotics and Automation*.

Scaramuzza, D., & Siegwart, R. (2008). Monocular omnidirectional visual odometry for outdoor ground vehicles. *Lecture Notes in Computer Science*, 5008, 206. doi:10.1109/TRO.2008.2004490.

Takasu, T. (2013). Rtklib: An open source program package for gnss positioning. URL: <http://www.rtklib.com/>.

Takasu, T., & Yasuda, A. (2009). Development of the low-cost rtk-gps receiver with an open source program package rtklib. In *International Symposium on GPS/GNSS, International Convention Center Jeju. Korea*. doi:10.1.1.472.7544.

Trucco, E., & Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR.

Zaman, M. (2007). High resolution relative localisation using two cameras. *Robot. Auton. Syst.*, 55, 685–692. URL: <http://portal.acm.org/citation.cfm?id=1284900.1284928>. doi:10.1016/j.robot.2007.05.008.

Zitova, B., & Flusser, J. (2003). Image registration methods: a survey. *Image and vision computing*, 21, 977–1000. doi:10.1.1.86.8364.

List of Figures

1	The test field showing GPS tracks of two test runs on Google Earth. Map data: Google, Lantmäteriet, Metria. The left-hand track represents the Field 1 experiment and the right-hand track the Field 3 experiment.	2
2	VO pipeline as presented in Scaramuzza & Fraundorfer (2011).	4
3	Distribution of distance to points: histogram of 10^6 points; maximum distance is limited to 10^6 m.	6

4	Experimental platform with sensor placement: FW-cam is the forward-facing stereo camera and DW-cam is the downward-facing stereo camera. The omnidirectional camera was not used in the research reported here.	8
5	Sample image from one of the forward-facing cameras in the simulated environment.	9
6	Sample images from stereo cameras: left- and right-hand images are placed side by side; resolution is 640×480 pixels.	9
7	Resulting 3D plot from using the Libviso algorithm on simulated KITTI 00 sequence data; black solid line = ground truth, red dashed line = visual odometry results, open circle = start position, solid circle = end position.	12
8	3D plot resulting from using the Libviso and Gantry algorithms on a simulated agricultural field; camera angle is 90° , open circle = start position, solid circle = end position.	14
9	Resulting 3D plot from using the Libviso algorithm with forward-facing cameras on Field 2, showing impact of different frame rates; open circle = start position, solid circle = end position.	17

List of Tables

1	Summary of related work using VO in an outdoor environment. "DOF" describes the degrees of freedom. "Method" specifies the motion estimation method, whether the camera is mono or stereo, and the direction the camera faces, i.e., forward (FW) or downward (DW). See explanation of the motion estimation techniques in Section 3.3.	3
2	Evaluated algorithms compared according to visual odometry pipeline steps.	6
3	Datasets from the agricultural field.	11

4	Visual odometry results of using the Libviso algorithm in simulation using the KITTI 00 sequence data for a total of 4540 frames. The error is measured using the KITTI method. Wrong motion estimates are the numbers of frames with translation error exceeding 1 m between two consecutive frames.	12
5	Visual odometry results for a simulated agricultural field; the Gantry method does not use multiframe optimization in these tests, and there are no motion estimation errors in these tests.	14
6	Comparison of the feature extractors and motion estimators: visual odometry results for a simulated agricultural field; the angle of the downward-facing cameras is 90°. . .	15
7	Results of field test on a fodder field.	16
8	Results of field test on a fodder field with a sub-sample of the data set. .	16

PAPER VI

LOCALIZATION OF AGRICULTURAL FIELD ROBOTS USING NATURAL LANDMARK AND VISION SLAM

S. Ericson and B. Åstrand (2016b). "Localization of agricultural field robots using natural landmark and vision SLAM.". Submitted to Sensors

Localization of agricultural field robots using natural landmark and vision SLAM.

Stefan Ericson ¹ and Björn Åstrand ²

¹ School of Engineering Science, University of Skövde, Skövde, Sweden; stefan.ericson@his.se

² School of Information Science, Computer and Electrical Engineering, Halmstad University, Halmstad, Sweden; bjorn.astrand@hh.se

* Correspondence: stefan.ericson@his.se; Tel.: +46 500 448509

† Current address: Affiliation 1

‡ These authors contributed equally to this work.

Academic Editor: -

Submitted to Sensors.

Abstract: Autonomous robots have to be able to estimate their position with high accuracy if they are to carry out advanced agricultural tasks in an open field environment. A vision-based system is recommended to complement the shortcomings of GPS solutions. In this paper we evaluate two visual SLAM algorithms, the feature-based ORB-SLAM and the dense LSD-SLAM on a challenging agricultural data set. The data set was acquired using a mobile robot equipped with a set of cameras and driven on an open agricultural field in a regular back and forth pattern. The resulting data set consists of four sequences with a path length of between 1.8 and 3.1 km. Both monocular and stereo versions of the algorithms are evaluated and LSD-SLAM is also applied on omnidirectional images. The results show that the stereo algorithms performed much better than the monocular version, and ORB-SLAM performed better than LSD-SLAM in three of the four sequences. The best result had 2.63 % translation error and 0.0321 deg m⁻¹ rotational error.

Keywords: Visual SLAM; Agricultural field robots.

1. Introduction

No pesticides may be used when growing organic vegetables, which makes weed control challenging. This problem is commonly solved by hand-picking the weeds, which is very labor intensive. However, this kind of task is particularly suited to automation. Mobile agricultural robots could autonomously navigate the fields and carry out different treatments. To be able to do this, a robot requires a reliable estimation of its position, where the degree of accuracy required dependent on the particular task. An accuracy of 10 mm is required for weed control using a precision tool to target individual plants [1]. This accuracy is required for both the tool's position and the crop identification sensor, unless the full pose of the robot is measured. A real-time kinematic GPS (RTK-GPS) can provide an absolute position measurement of this accuracy at best performance, but it can also lose accuracy very quickly if the number of visible satellites decreases. Moreover, a GPS only provides a measurement of the position in 3D, rather than a full 6 degrees of freedom (6-DoF) pose. In addition, a map is required to interpret the position measurement from a GPS. It seems that a vision-based system can both achieve more degrees of freedom and has the ability to build a map.

Visual odometry is a technique that can be used to estimate the camera pose from a sequence of images. It has been used for localization both in urban [2,3] and agricultural environments [4,5]. In visual odometry feature points are extracted, matched and tracked between consecutive images, and the camera's motion is estimated from the relative motion of the feature points. The position is then derived by summing the relative motions from the known starting position. Given a calibrated

stereo camera, up to 6-DoF can be estimated. The best result on the KITTI vision benchmark list [6] for visual odometry has a 0.88 % position error and a $0.0022 \text{ deg m}^{-1}$ rotational error [7].

The major drawback of visual odometry is that errors accumulate. My earlier work shows that when using visual odometry in an open agricultural field, the data suffers from errors due to the uneven point distribution of feature points [4]. Uneven point distribution, combined with high camera frame rates and error accumulation, gives poor results.

A solution to the error accumulation problem is to detect some kind of landmark with a known position in the environment. When the robot detects such a landmark, the position can be corrected and the accumulated error reduced. The use of landmarks is demonstrated in [8], where four artificial landmarks were placed at the corners of a $50 \times 50 \text{ m}$ square on an agricultural field. These landmarks were detected using an omnidirectional camera, which provides angles to each landmark, so that the position can be estimated in 2D. Their result showed a mean position error of approximately 0.25 m.

A more general solution to the localization problem is not to use artificial landmarks at known, calibrated positions but to use natural landmarks available in the scene. That requires a technique that can both localize the robot and build a map with landmarks, a technique that is usually referred to as simultaneous localization and mapping (SLAM). This is a widely explored research field in which the algorithms can be divided into three paradigms based on their solution method: Kalman filter approaches, particle filter approaches, and graph-based approaches [9].

The Kalman filter approaches usually use an extended Kalman filter (EKF) to model the robot pose and the position of all landmarks. Thus the covariance matrix includes both pose and all landmarks, and it grows in size as new landmarks are found. This leads to a system that is computationally intensive, which makes this approach suitable only for systems with up to 1000 landmarks. The requirement for EKF-SLAM is that the noise model is Gaussian and that the positions of the landmarks can be measured from a single position, that is, both range and bearing are known. For visual SLAM (VSLAM) it is possible to use a stereo camera setup. With proper initialization of the initial state, VSLAM can also be used with monocular cameras.

In particle filter SLAM, the states are represented by samples from the probability distribution, that is, as particles. The particles are weighted based on their importance, and then resampled. The advantages of this approach are that the particle filter is not limited to Gaussian distribution and supports multiple hypotheses regarding the pose. An effective implementation of particle filter-based SLAM is the fastSLAM [10,11]. Two drawbacks of the particle filter SLAM are its linearization of the non-linear motion model, which is not optimal, and the particle depletion phenomenon, where the system is over-confident on the basis of false hypotheses.

In the graph-based approach, every pose of the robot and the landmarks is represented by nodes on a graph. The relationships between these poses are represented by edges, which are seen as soft constraints. The system can be solved by minimizing the energy of the net [12]. There are many approaches to finding the optimal solution, and numerous methods are presented in literature [13]. The main advantage of this approach is that the computational complexity is linear with the number of nodes, i.e. $\mathcal{O}(N)$, compared to EKF-SLAM in which the complexity is $\mathcal{O}(N^2)$. Another advantage over EKF-SLAM is that graph-based SLAM can also handle outliers.

Examples of agricultural robots that use a SLAM approach are found in [14–16]. In all these examples, the agricultural scene consists of rows of plants with visible stems. The robots move between or around the rows. In [14] the landmarks are obtained both from laser scans and from one monocular camera detecting stems. The use of two different sensors for landmark detection increased the robustness by rejecting false matches. Their result shows an error of less than 0.5 m when navigating around a $70 \times 70 \text{ m}$ field. Others use a particle filter approach to solve the SLAM problem in a maize field [15]. The stems were detected using a laser scanner; the RMS-error of the position obtained was 0.04 m and 2.4 deg. In [16] the only sensor is a laser scanner. Their SLAM approach does not fit any of the above-mentioned techniques. Instead they transformed the scan into an image, and then used image registration techniques to find a match between current and previous

scans. Thus a map and the robot pose in 3-DoF were obtained. The reported result is $0.16\% \pm 0.1\%$ on a total path of 252 m in an apple orchard and in a vineyard. In all the approaches reported above, 2D laser scanners were used to detect the plant stems. However, such scanners are not suitable for use in contexts where the plants lack a distinctive stem, such as in open fields where fodder or oats are grown.

Another approach is to base the SLAM algorithm on data captured by a camera using visual SLAM (VSLAM). Most VSLAM methods use a feature-based approach, with features detected in the image being used as landmarks. The algorithm PTAM (parallel tracking and mapping) is one example. It was developed for artificial reality with the goal of including virtual 3D objects in the camera view [17]. It requires that the camera position be related to a map of objects, and hence it shares the same problem as SLAM. The code consists of two threads, with one thread tracking the camera position and the other building the map. Due to the different applications, there are some differences between a PTAM and a SLAM system operating in a large-scale environment. PTAM is limited to small-scale operations and lacks functionality such as loop closure, handling of occlusions, and low invariance to viewpoint of the relocation [18].

ORB-SLAM [18] is another example of a feature-based VSLAM algorithm that is built upon PTAM, but it uses ORB features [19] instead of PTAM's FAST corners. It handles the shortcomings of the PTAM algorithm by including place recognition, loop closing, and covisibility information, which will ensure operation on a large scale. Place recognition is based on visual words, and hence a vocabulary has to be generated and included for the specific environment. The algorithm consists of three threads, two of which are similar to PTAM's tracking and mapping, while the third is for detecting and closing loops. ORB-SLAM uses graph-based SLAM to optimize the location of keyframes in the same way as PTAM. ORB-SLAM also appears on the KITTI benchmark suite, with reported errors of 1.15 % for position and $0.0027 \text{ deg m}^{-1}$ for rotation. An important observation is that the algorithm actually failed on one sequence from a highway where there were too few trackable features. That particular sequence is very similar to an open agricultural field.

The main drawback of using a feature-based approach is that only information that conforms to the feature type can be used [20]. Another option is to use a dense approach, which uses all information in the image. One example is large-scale direct monocular SLAM (LSD-SLAM), which aligns images directly and uses a filter-based estimation to create semi-dense depth maps [20]. A graph-based SLAM approach is used in which the vertices represent keyframes, and the edges a 3D similarity transform. Thus corrections can be made for both accumulated error and scale drift. This work has been extended to work with stereo cameras [21] and omnidirectional cameras [22]. Their result on the KITTI benchmark is a 1.20 % translation error and $0.0033 \text{ deg m}^{-1}$ rotational error.

In this paper we evaluate two competing SLAM algorithms, the feature-based ORB-SLAM and the semi-dense LSD-SLAM, on challenging agricultural data sets. The algorithms are applied using a monocular forward-facing camera and also a stereo forward-facing camera. One of the algorithms was also applied using an omnidirectional camera with a catadioptric lens. This paper thus contributes to the qualitative evaluation of the accuracy and robustness of the localization problem in an open agricultural field environment using vision-based SLAM.

2. SLAM algorithms

The two competing algorithms that are evaluated are ORB-SLAM [18] and LSD-SLAM [21]. For clarity and readability, a summary of each algorithm is given in Section 2.1 and Section 2.2. In Section 2.3 the stereo variants of both algorithms are presented, and finally the SLAM using omnidirectional cameras is presented in Section 2.4.

2.1. ORB-SLAM on a monocular camera

The ORB-SLAM algorithm is a feature-based approach that was designed from scratch. It is based on the PTAM [17] algorithm, but with improvements to better suit the SLAM problem [18].

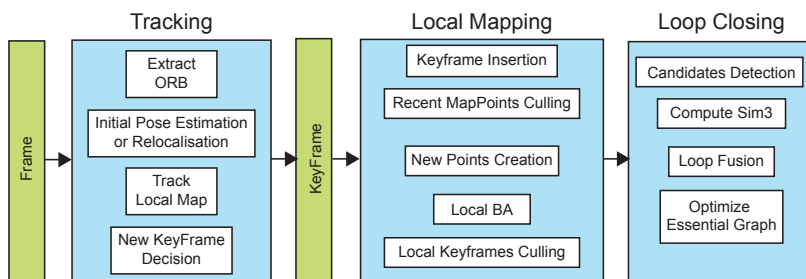


Figure 1. ORB-SLAM system overview [18]. The code is divided into three threads that are run in parallel: tracking, local mapping, and loop closing. The tracking thread extracts and matches ORB features over consecutive frames and decides whether a new keyframe is to be created. Local mapping utilizes a pose graph of keyframes and bundle adjustments to minimize the drift in position estimation. The loop closing thread identifies loops by using place recognition with bags of words and a second pose graph.

The algorithm consists of three threads that run in parallel: tracking, local mapping, and loop closing. Figure 1 shows a block diagram of the algorithm.

The main concept of the algorithm is to use the same feature for tracking, mapping, and place recognition. ORB features [19] are used, which are a fast alternative to SURF and SIFT features. Such features are basically oriented multi-scale FAST corners with a 256-bit binary descriptor to enable fast detection and matching. They are much more invariant to different viewpoints than the PTAM algorithm.

The tracking thread starts by extracting features in the image, which are tracked on a local map. Constant motion of the camera is assumed to predict the track, which is then refined using feature matching. If there is no pose estimation from the last frame, or if the tracking is lost, the local map has to be reinitialized using place recognition. The tracking thread also decides when to create a new keyframe. Tracking performance and the number of frames since the last keyframe are used as criteria to insert a new keyframe. The policy on inserting keyframes is generous since the local mapping thread will discard redundant keyframes.

The local mapping thread optimizes the keyframe pose based on a covisibility graph. This graph is created by adding the keyframe poses as nodes, with each edge representing the number of features that are visible in the previous and current keyframes. This is used for weighting when optimizing the pose graph. All feature points are included in the keyframe during its creation, even if they are considered to be outliers in the tracking. The matching of features between keyframes determines which feature points to keep. Only the strongest feature points are kept as map points. The local mapping thread also determines whether keyframes are redundant by evaluating overlapping map points and point accuracy. Keyframes in which 90 % of the map points are visible in more than three other keyframes are removed.

The third thread's task is to detect loop closure and perform relocalization. Loops are detected by recognizing previous places using bags of words [23]. A visual vocabulary is created offline using ORB descriptors of a large set of images. This vocabulary is implemented in a hierarchical tree structure that enables fast conversion of images to bag-of-word vectors. Each keyframe is described by a bag-of-word vector that is matched for similarity to detect loop closures. The same method is used to recover after a tracking failure. Once a loop closure is detected, another pose graph optimization is performed to correct any accumulated error. This pose graph is referred to as an

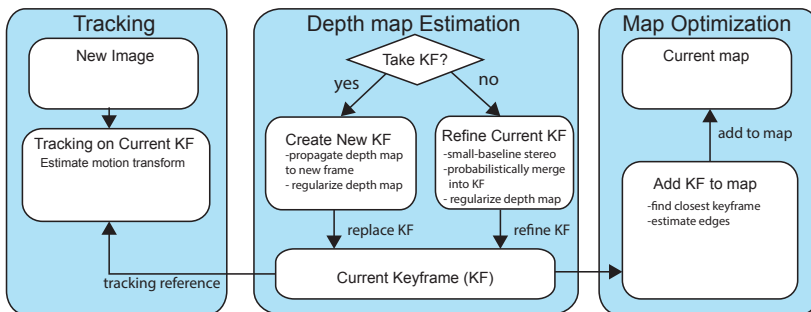


Figure 2. LSD-SLAM system overview [21]. This algorithm consists of three major components: Tracking, depth map estimation, and map optimization. The tracking estimates relative motion by minimizing the difference between the previous and current keyframes. The depth map estimation determines whether a new keyframe is to be created, and the corresponding depth map is either refined or initialized. The map optimization utilizes a pose graph of keyframes to optimize keyframe positions.

essential map. It is a map of keyframes similar to the covisibility map, but there are fewer edges in order to make it efficient and robust.

The ORB-SLAM algorithm requires that the map be initialized with a set of points. This initialization uses both homography, which assumes a planar scene, and a fundamental matrix that also covers nonplanar scenes. The method that provides the least error is selected. The limits are quite restrictive, so that initialization only occurs when the error is small. However, this might lead to no initialization at all if the camera follows an adverse path.

2.2. LSD-SLAM on a monocular camera

LSD-SLAM consists of three parts: tracking, depth map estimation, and map optimization. The tracking procedure tracks changes in the image relative to the previous keyframe to estimate camera motion. The depth map estimation is performed on the keyframes and uses filtered contributions from many small stereo comparisons. Finally a global map is built and optimized by adding the keyframe's depth map. Loop closures and scale drift are detected by applying the same tracking technique as between consecutive images. A block diagram of the algorithm is shown in Figure 2.

The tracking procedure uses direct alignment between images to find the motion of the camera. This is done by minimizing the photometric error between a reference image and a transformed version of the current image. This 3D similarity transform contains translation, rotation, and scaling according to Equation 1.

$$S = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (1)$$

where $\mathbf{R}$ is 3x3 rotation matrix, $\mathbf{t}$ is 1x3 translation vector and s is scale. A minimal representation of this transform expressed in Lie algebra is $\xi \in \mathfrak{sim}(3)$, which corresponds to $\xi \in \mathbb{R}^7$. The additional degree of freedom is the scale factor.

The photometric error is minimized using a weighted Gauss-Newton optimizer that gives a maximum-likelihood estimator of the transform ξ . Different weighting schemes are applied to minimize the impact of outliers.

Finally, the uncertainty is approximated using a Gaussian residual assumption and error propagation using the Jacobian according to Eq 2.

$$\Sigma_f \approx J_f \Sigma_X J_f^T \quad (2)$$

where $f(X)$ is the output function of the input X , Σ_X is the covariance of the input, and J_f is the Jacobian of f .

The depth map is created when the distance or the rotation relative to the previous keyframe is too large. Then a new keyframe is created by projecting points from the previous keyframe. Outliers are removed and the scale is normalized. The keyframe's depth map is then refined by tracking new frames, which will give additional stereo comparisons. All these contributions are fused together in the depth map using a filtering approach.

Scale drift is a major issue in monocular SLAM. Since the scale is not observable, the algorithms can only estimate the pose up to a scale. Over a long path, the scale will drift, which is a major source of error. The solution is to normalize all depth maps to a mean distance of one, and then incorporate the scale in the motion transform estimation ξ . In that way the algorithm handles both loop closures and scale drift.

The last step is to optimize the location of the keyframes using pose graph optimization. The pose of the keyframe is represented by the nodes and the transform between frames is represented by edges. The framework g2o [24] is used to perform the non-linear optimization of the error function.

Monocular LSD-SLAM requires the depth map to be initialized. In the original algorithm this is done using random depths with large variance. This initialization requires the camera to move in the direction of the image plane so many stereo comparisons can be made to build a valid depth. This motion is not possible for forward-facing cameras mounted on an agricultural vehicle with Ackerman steering, and so the depth map has to be initialized in another way. In this work we have modified the initialization and use the assumption that the scene represents an open agricultural field that can be modeled as a plane with the cameras facing forward. Then the depth map can be initialized based on the plane's projection onto the image plane, that is, pixel location in the image, so that points at the top of the image are far away and points at the bottom of the image are close. Let (u, v) represent pixel coordinates in the image, h image height, and $d(u, v)$ the depth in that pixel. Then the depth map can be calculated according to Equation 3.

$$d(u, v) = 1 - v/h \quad (3)$$

The ORB-SLAM also requires the depth to be initialized. The method used to initialize LSD-SLAM is applied to ORB-SLAM, but the algorithm failed to initialize correctly. Instead an alternative method is used, where the algorithm is forced to initialize by adding a stereo image for the first frame and then performing regular stereo matching.

2.3. SLAM on stereo camera

The stereo extension of LSD-SLAM adds one step into the keyframe depth map estimation [21]. The additional step uses the fixed baseline of the stereo camera to estimate the depth. The result is merged into the keyframe's depth map similar to the refinement contribution of each frame. This removes scale as a variable from the pose vector ξ . The ability to handle illumination differences is an improvement compared to the original algorithm. This is implemented by modifying the photometric residuals to be invariant to affine light changes [21].

2.4. SLAM on omnidirectional camera

SLAM with an omnidirectional camera is a variant of monocular SLAM using a different camera model [22]. There are many different models for cameras with fisheye or catadioptric lenses. One common assumption is that there is a single effective viewpoint, that is, there is a single point in

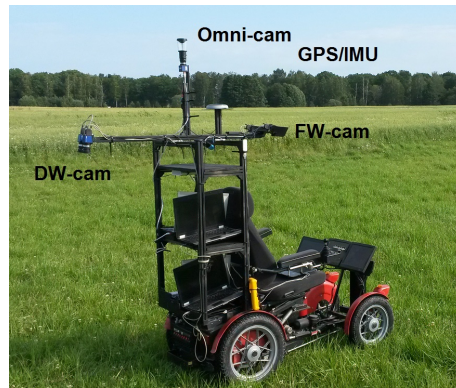


Figure 3. Mobile robot used for the experiments. DW-cam is the downward-facing stereo camera, FW-cam is the forward-facing stereo camera, and Omni-cam is the omnidirectional camera. The GPS/IMU is used as ground truth in the experiment and consists of an RTK-GPS and a combined 3-DOF magnetometer, accelerometer, and gyro (9-DOF Razor).

space through which all light rays pass. This assumption holds true for most high-end lenses. A requirement on the camera model set by the LSD-SLAM algorithm is that there be a closed form unprojection function. This disqualifies camera models that use Taylor expansions [25]. Instead a unified model is presented that uses two projections, first onto a unit-sphere and then onto an image plane located along the z-axis at distance $-\xi$. Let (x, y, z) be a point in space, and (u, v) the pixel coordinates in the image. The projection of a point in space can be calculated according to Equation 4.

$$\pi_u(x) = \begin{bmatrix} f_x \frac{x}{z + \|z\|\xi} \\ f_y \frac{y}{z + \|z\|\xi} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \quad (4)$$

The corresponding unprojection function is shown in Equation 5.

$$\pi_u(u, d)^{-1} = \frac{1}{d} \left(\frac{\xi + \sqrt{1 + (1 - \xi^2)(\tilde{u}^2 + \tilde{v}^2)}}{\tilde{u}^2 + \tilde{v}^2 + 1} \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \xi \end{bmatrix} \right) \quad (5)$$

where

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} (u - c_x)/f_x \\ (v - c_y)/f_y \end{bmatrix} \quad (6)$$

and (f_x, f_y) is the focal length, and (c_x, c_y) is the principal point.

3. Experiments

The algorithms were evaluated on a data set with four sequences acquired by a mobile robot on an agricultural field. There were three sets of cameras: a forward-facing stereo camera, a downward-facing stereo camera, and a monocular omnidirectional camera. There was also an RTK-GPS and an inertial measurement unit (IMU) that was used as ground truth. The robot was driven manually in a back and forth pattern with headland turns on two different fields.

3.1. Mobile robot

The mobile robot and its sensors are shown in Figure 3. The stereo cameras consist of two mvBlueFox-120a (Matrix Vision, Oppenweiler, Germany), which are configured to capture 640×480 -pixel gray-scale images at 20 Hz. The omnidirectional camera is a mvBlueCougar-X (Matrix Vision, Oppenweiler, Germany) with a catadioptric lens VS-C450MR-TK (KumoTek Robotics, Richardson, Texas, USA). This camera is configured to acquire 1200×1200 -pixel gray-scale images at 2 Hz. The downward-facing camera is mounted 0.9 m behind the omnidirectional camera and has a baseline of 0.05 m. The cameras are tilted slightly backwards to exclude any visible parts of the robot. The forward-facing cameras have a baseline of 0.2 m and are mounted with a slight downward tilt in order to include as much of the ground as possible but still keep the horizon and the treeline within the images. There are sun covers on the forward-facing camera to minimize sun flares that appear when driving towards the sun.

The RTK-GPS system consists of two LEA-4T GPS receivers (u-blox, Thalwil, Switzerland) with high performance RTK antennas. One receiver is used as a base station and is mounted at a fixed position within 1 km of the test field. The other unit (rover unit) is mounted on the robot. These are linked using mobile internet. *Librtk* [26,27] software is used to obtain position estimation with a centimeter accuracy. The IMU, a 9-DOF Razor (sparkfun, Niwot, CO, USA) is used to obtain the rotation of the robot. The IMU and the GPS antenna are as close as possible to each other, and both are aligned with the rear wheel axis. The omnidirectional camera has to be mounted above the GPS antenna to avoid an interfering antenna cable being in view of the camera. The best position for the camera was evaluated experimentally to minimize its impact on the GPS signal-to-noise ratio (SNR). A lower SNR might result in a satellite being rejected from the position estimation, leading to lower precision of the GPS result.

There are five different computers on the robot that acquire data, one for each stereo camera, one for the omnidirectional camera, one for the RTK-GPS, and finally one used as a time server. All the sensors are triggered by hardware, and the time server measures the trigger pulses to provide accurate timestamps for each pulse. All data are stored on disks and analyzed offline. Figure 4 shows sample images of the forward- and downward-facing cameras, while Figure 5 shows an example of an image captured by the omnidirectional camera.

3.2. Calibration and image preprocessing

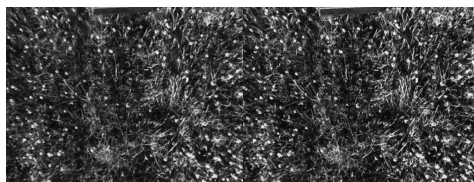
The cameras were calibrated using standard procedures and appropriate Matlab toolboxes. Two chessboards were used, both with 8×6 squares, but of different sizes. The smaller chessboard (30 mm squares) was used for the downward-facing camera, and the larger chessboard (100 mm squares) was used when calibrating the forward-facing cameras and the omnidirectional camera. An image sequence was recorded with the pattern shown in different positions so that the entire field of view was covered. The stereo cameras were calibrated using Bouguet's toolbox [28] and the images were undistorted and rectified before they were used by the SLAM algorithms. The omnidirectional camera is calibrated using Mei's toolbox [29].

3.3. Agricultural field dataset

The experiments were conducted in 2014 and 2015 on two different fields. One was a field of oats with a plant height between 20 and 500 mm, and the other a fodder field consisting mainly of clover, grass, and dandelions. The average plant height of this field was estimated to be 200 mm. Four sequences in total were captured on these fields, one on the oat field and three on the fodder field. The robot was controlled manually in a common back and forth pattern with headland turns. The robot was stopped for a short while after every second headland turn. Figure 6 shows the trajectory on the field of two sequences taken from the ground truth.



(a) Sample images from the forward-facing stereo camera.



(b) Sample images from the downward-facing stereo camera.

Figure 4. Sample images from stereo cameras: left- and right-hand images are placed side by side; resolution 640×480 pixels.

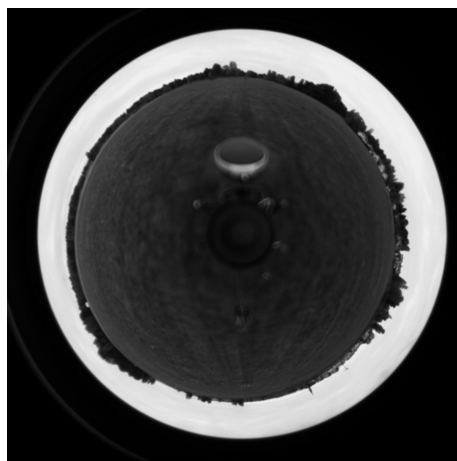


Figure 5. Sample images from omnidirectional camera, resolution 1200×1200 pixels. The GPS antenna is the circular disk placed ahead of the omnidirectional camera. This is the part of the vehicle that is the most visible. The downward-facing cameras are also visible in the opposite direction, but these are less dominant due to their small size and low image resolution.



Figure 6. The test field showing GPS tracks of three test runs on Google Earth. Map data: Google, Lantmäteriet, Metria. The left-hand track in white represents the Field 1 experiment, the right-hand track in white the Field 2 experiment, the red track is the path of the Field 4 experiment.

The weather during the experiments varied, and there was light rain during the sequence on the oat field. This resulted in images with lower contrast compared to the other sequences. There was also one rain drop that hit the lens of the left forward-facing camera and stayed there for half the test. The second test on the fodder field was done on the evening of a sunny day. This resulted in shadows from the robot and sun blur interfering with the forward-facing camera. The last two sequences were obtained in partly cloudy weather, which resulted in rapid changes of contrast between the sunny and cloudy parts. In sequence 3, three curious women approached the robot and stopped for five minutes to chat with the operator.

The conditions during the experiments may not have been optimal for visual localization, but were highly representative of an agricultural scenario.

A summary of the sequences is shown in Table 1.

Table 1. Datasets from the agricultural field.

Sequence label	Year	Field type	Weather	Distance	Frames	Max. speed (mm frame ⁻¹)	Max. rotation (deg frame ⁻¹)
1	2014	Oat	rainy	2093	32700	109	0.158
2	2014	Fodder	sunny	1838	33400	84	0.196
3	2015	Fodder	partly cloudy	2366	58380	66	0.151
4	2015	Fodder	partly cloudy	3065	63000	78	0.144

3.4. Error calculations

The translational and rotational errors were calculated in the same way as the odometry evaluation in the KITTI benchmark suite [6] for all subsequences of length 100, 200, ..., 800 meters. The translation error is the average end error of all sequences and includes error in the x , y , and z directions. The rotational error includes errors in terms of roll, pitch, and yaw, and is calculated as the average heading error at the end position of each subsequence. This error is also divided by the traveled distance, giving a measure in deg m⁻¹.

4. Results and Discussion

All the algorithms tested were first validated on data sets where the result was known. For the monocular and stereo variants of ORB-SLAM and LSD-SLAM, the KITTI Benchmark data set was used [6]. The omnidirectional algorithm was tested on a data set for omnidirectional SLAM¹.

The results of the monocular algorithms show that both ORB-SLAM and LSD-SLAM fail to correctly estimate the full path of the field tests. Figure 7 shows a part of the result when applying ORB-SLAM on the Field 2 data set. The algorithms manage to correctly estimate the position (up to a scale) for the first straight path. During the turn they fail to estimate the depth of the new points correctly, and the translation between frames is underestimated. This underestimation continues on the way back, until the uncertainty of the feature points on the field is so large that these are removed. This results in feature points only at the horizon, which gives little information on camera translation. No new keyframes are introduced since the horizon appears the same, which produces a large jump in the position plot. The rotation is, however, estimated correctly. After the next turn one of two errors occurs. Either the track of these points is lost, which results in an unrecoverable error, or the algorithm continues with large-scale drift as result (Figure 8). Similar behavior occurred on the other agricultural sequences.

The result for monocular LSD-SLAM applied to the same data set shows that it is very hard to make it initialize and start correctly. The most common behavior is that it estimates the first frames as being in a location ahead of the current location. Then it fails to detect motion forward and instead jumps around at the same spot. The algorithm did produce a good result for one long sequence where everything worked as expected, but this result was not reproducible due to the algorithm's stochastic behavior. Figure 9 shows the 3D point cloud and a depth map for one of the keyframes.

The stereo variants of the algorithms work much better, and all sequences can be tracked over each path's full length. The result is shown in Table 2, and individual plots for the different sequences are shown in Figure 10 for ORB-SLAM and Figure 11 for LSD-SLAM. Due to the randomness of the algorithms, the median result of five test runs is shown in the figures. Minimum, median, and maximum values are reported in Table 2. ORB-SLAM shows very good results on the Field 3 sequence, in which the 3D end position is misaligned by 7.1 m after a 2.4 km run. There is no loop detected in any run, even though both Field 3 and Field 4 include one loop closure at the end of the run. When first looking at the images, one can easily believe that there should be loop closures at every second headland turn. Indeed these images look very similar, but in fact the only object that is the same is the horizon. The rest of the image contains a new part of the field that was not seen before. That means the algorithm failed to map the features on the horizon with the features from the previous row, which leads to a drift in the heading angle. This becomes very clear in the Field 4 sequence, where it shows a result similar to pure visual odometry.

Stereo LSD-SLAM yields a result that is less robust than ORB-SLAM. As in the monocular case, it occasionally makes a false match with the previous keyframe and temporarily loses track. This leads to a large error that decreases the overall performance. One example can be seen in the Field 1 sequence, where the heading is turned upwards. In one sequence (Field 4) it provides a slightly better result than ORB-SLAM, mostly due to lower drift in the heading. Both algorithms show stochastic behavior, and the result shows that the variance is larger using LSD-SLAM.

These stereo algorithms are applied on the images from the downward-facing cameras. The result shows that ORB-SLAM could estimate the path, but there is a large drift in pitch estimation. The LSD-slam also shows large drift, but in all rotations, leading to an even bigger error. There is no major advantage to use SLAM approach compared to an visual odometry approach on the downward-facing cameras, since the overlap is only between a few consecutive frames.

¹ Datasets - SLAM for Omnidirectional Cameras <http://vision.in.tum.de/data/datasets/omni-lsdslam>

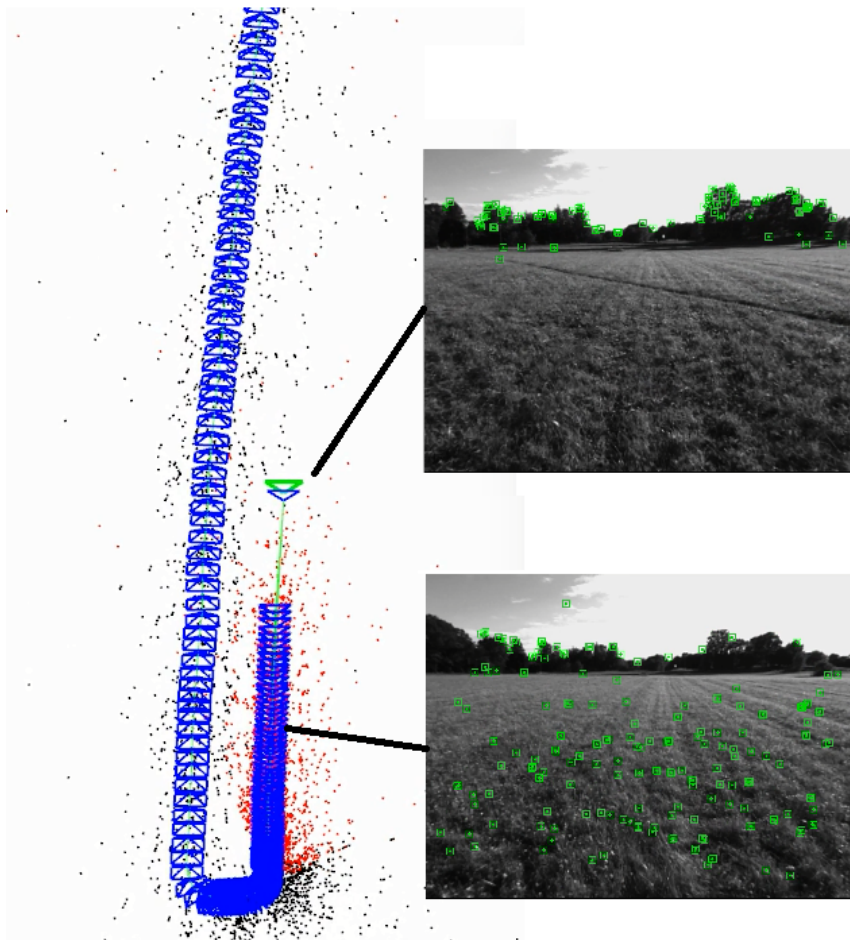


Figure 7. ORB-SLAM on monocular forward-facing camera applied to the Field 2 sequence. The blue marks show the pose (position and heading) of each keyframe. The black dots show detected feature points that were previously used, i.e. the map. The red dots are the feature points that are currently used in the local map. Starting from the top left in the image, the position is correctly estimated until the turn. During the turn more keyframes are introduced indicating many new feature points. On its way back the scale drifts due to difficulties in estimating the translation correctly. At the end of the second straight path, all feature points on the field are lost and the algorithm estimates translation with low accuracy. No new keyframes are introduced since the appearance of the horizon is the same, which leads to a large jump in the position.

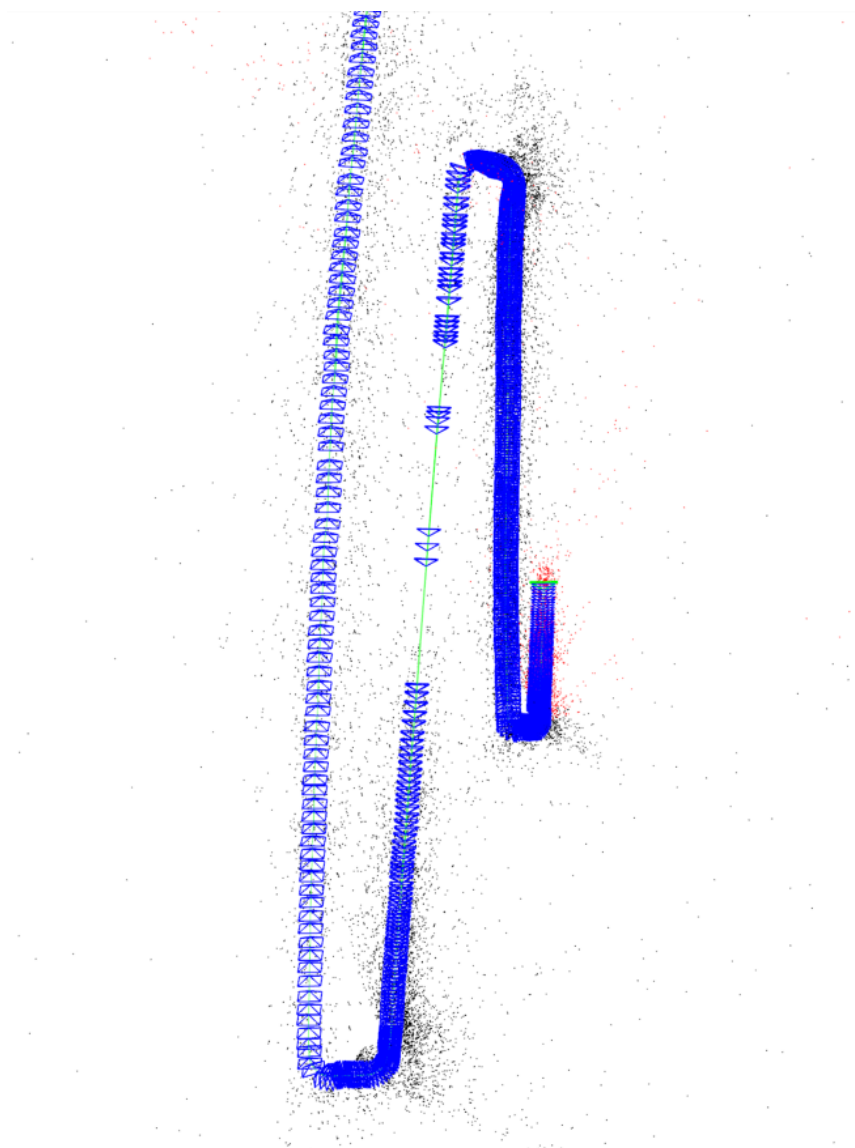


Figure 8. ORB-SLAM using the monocular forward-facing camera applied to the Field 2 sequence. The blue marks show the pose (position and heading) of each keyframe. The black dots show detected feature points that were used previously, i.e. the map. The red dots are the feature points that are currently used in the local map. This plot shows scale drift although the algorithm does not fail during the second headland turn.

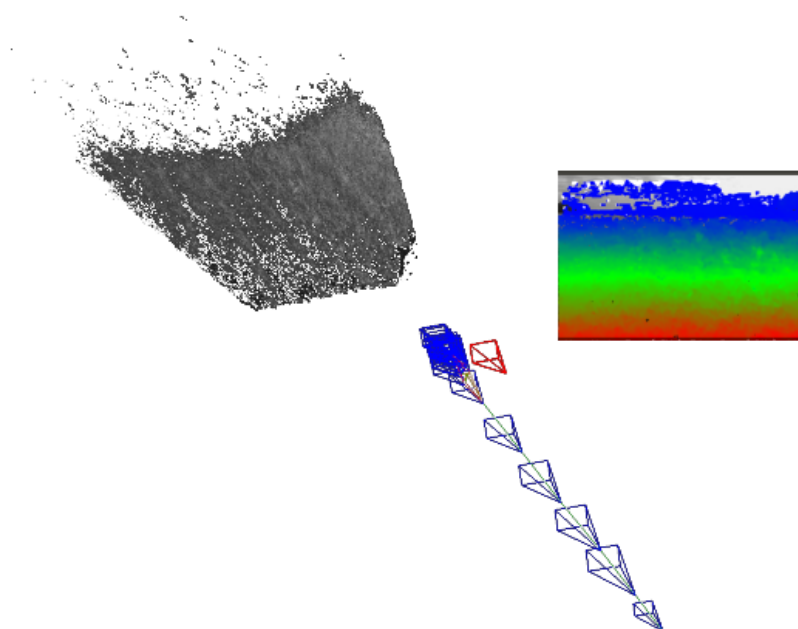


Figure 9. LSD-SLAM using the monocular forward-facing camera applied on Field 2 sequence. The blue marks show the keyframe position, the black dots show the estimated 3D point cloud, and the image shows the depth map after initialization. This figure shows a failure mode in which the algorithm fails to detect translation. The failure causes the position to become more or less stuck, and new frames contribute to a point cloud at the same position as the previous one.

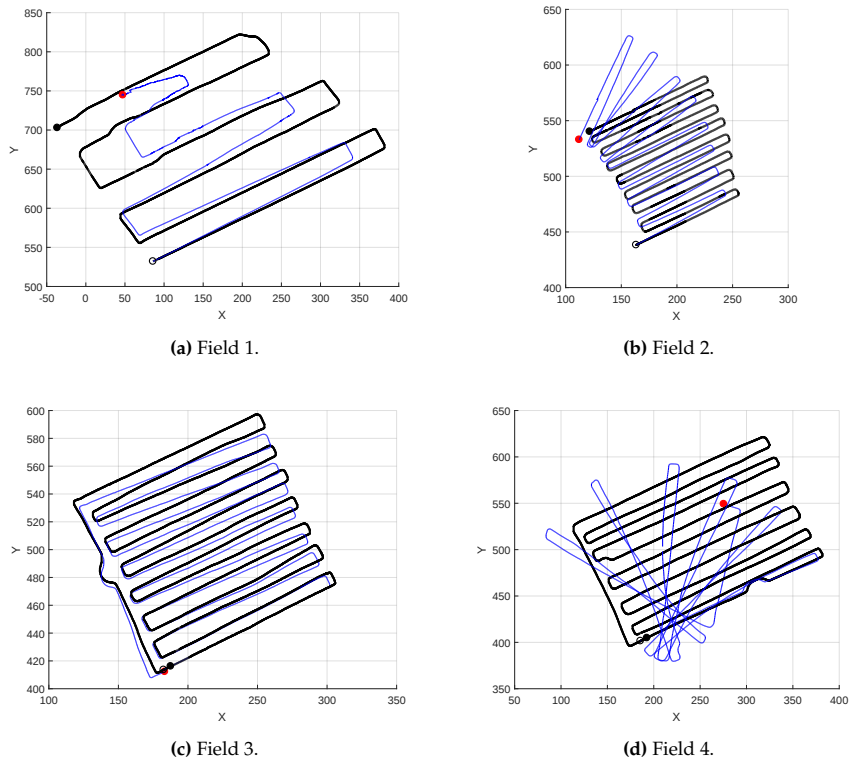


Figure 10. Resulting plot from using the ORB-SLAM stereo algorithm with forward-facing cameras; black line = Ground truth, blue line = SLAM result, open circle = start position, solid circle = end position. Field 1 test shows a rather large drift in tilt, and the translation is underestimated. This is due to lower contrast in this data set. Field 2 shows a fair result but suffers from drift in heading, especially at the end of the run. Field 3 provides the best result showing a low error in both translation and rotation. There are, however, no loop closures and there is still error accumulation. In Field 4 the heading estimation fails leading to a large rotation error.

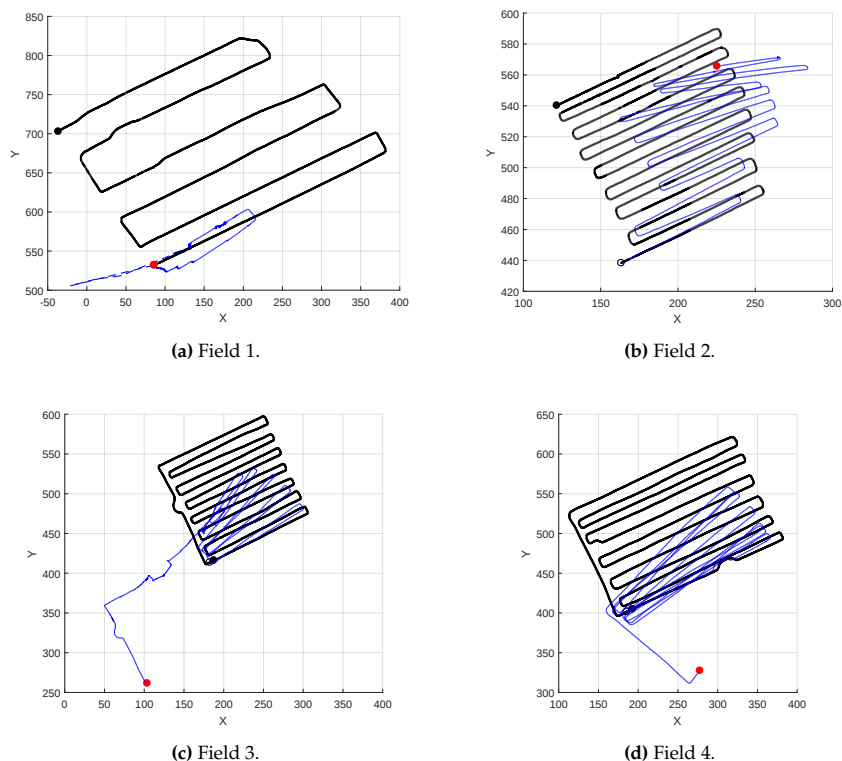


Figure 11. Resulting plot from using the LSD-SLAM stereo algorithm with forward-facing cameras; black line = Ground truth, blue line = SLAM result, open circle = start position, solid circle = end position. Field 1 provides a very poor result. The rotation estimation fails leading to a translation in z-direction, which is not visible in the plot. Field 2 shows less drift in rotation, but still too much to keep the path in the same plane. In Field 3 test some headland turns are missed. Field 4 shows better performance than ORB-SLAM and the main error is the drift in heading.

Table 2. Results of field test using stereo cameras. Median of five runs.

Test setup	Distance (m)	Measure	ORB-SLAM stereo		LSD-SLAM stereo	
			Translation error	Rotation error (deg m ⁻¹)	Translation error	Rotation error (deg m ⁻¹)
Field 1	2093	min	14.9%	0.0266	37.2%	0.2243
		median	15.0%	0.0264	39.5%	0.2772
		max	15.3%	0.0266	55.3%	0.2168
Field 2	1838	min	3.83%	0.0454	6.77%	0.0516
		median	3.92%	0.0487	8.08%	0.0663
		max	4.03%	0.0491	12.9%	0.0542
Field 3	2366	min	2.55%	0.0322	11.7%	0.0436
		median	2.63%	0.0321	12.6%	0.0459
		max	2.68%	0.0338	13.7%	0.0480
Field 4	3065	min	6.21%	0.0545	4.61%	0.0414
		median	6.97%	0.0631	5.03%	0.0380
		max	7.14%	0.0642	5.24%	0.0379

The result using LSD-SLAM with the omnidirectional camera shows that it is hard to estimate translation. In contrast to the fisheye lens used in [21], the catadioptric lens has a large unusable region due to the central obstruction of the lens and the fact that the vehicle itself is visible. This region does not represent any motion and should not be used for estimating motion. Figure 12 shows a depth map of LSD-SLAM using the omnidirectional camera. It shows that the horizon only provides a texture that is trackable between frames and where depth can be estimated.

5. Conclusions

In this paper we evaluated two visual SLAM algorithms, the feature-based ORB-SLAM and the dense LSD-SLAM, on a challenging agricultural data set. The data set was acquired using a mobile robot equipped with a set of cameras and driven on an open agricultural field in a regular back and forth pattern. The resulting data set consists of four sequences with a path length varying from 1.8 to 3.1 km. Only two of the sequences included loop closure, both once at the end of the sequence. The results show that the stereo algorithms produced much better results than the monocular variant of the algorithms. Apart from overcoming the scale ambiguity, the stereo algorithm provides much more robust results and did not require additional assumptions for initialization. ORB-SLAM performed better than LSD-SLAM, even though the full potential of SLAM with loop closure did not appear, since both methods failed to detect the loop closures. The result then becomes more like visual odometry with an accumulating error. The result using the omnidirectional camera showed that the algorithm failed to estimate the translation, which is a consequence of not being able to detect points close to the camera. Instead only the horizon can be detected and tracked, which provides only an estimate of the robot's orientation. Further work is required to modify the ORB-SLAM algorithm to enable loop closure for orientation at each headland turn. That could possibly be done if the algorithm were to create two types of keyframes to separate feature points detected far away from those detected closer to the robot.

Bibliography

1. Lechner, W.; Baumann, S. Global navigation satellite systems. *Computers and Electronics in Agriculture* **2000**, *25*, 67–85.
2. Nistér, D.; Naroditsky, O.; Bergen, J. Visual Odometry. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2004**, *1*, 652–659.

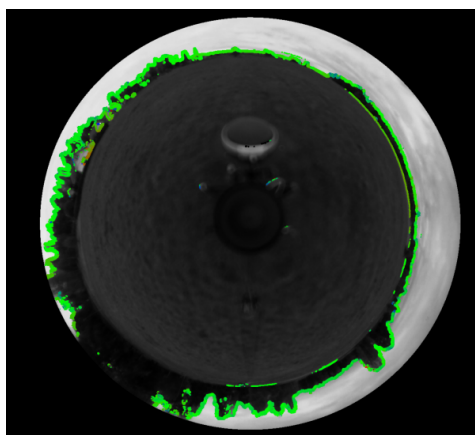


Figure 12. A sample depth map from LSD-SLAM applied to the omnidirectional camera. Only points in the horizon are matched and used by the algorithm.

3. Geiger, A.; Ziegler, J.; Stiller, C. StereoScan: Dense 3d Reconstruction in Real-time. *IEEE Intelligent Vehicles Symposium*, 2011; pp. 963–968.
4. Ericson, S.; Åstrand, B. Ego-motion estimation by an agricultural field robot using visual odometry. *Submitted to Biosystems engineering* **2016**.
5. Jiang, D.; Yang, L.; Li, D.; Gao, F.; Tian, L.; Li, L. Development of a 3D ego-motion estimation system for an autonomous agricultural vehicle. *Biosystems Engineering* **2014**, *121*, 150–159.
6. Geiger, A. The KITTI Vision Benchmark Suite. online, 2016.
7. Cvišić, I.; Petrović, I. Stereo odometry based on careful feature selection and tracking. *European Conference on Mobile Robots* 2015, 2015.
8. Li, M.; Liu, Z.H.; Huang, J.; Dai, S.; Wakabayashi, K.; Imou, K. Artificial Landmark Positioning System Using Omnidirectional Vision for Agricultural Vehicle Navigation. *Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on*. IEEE, 2012, pp. 665–669.
9. Stachniss, C. Lecture notes Robot Mapping - WS 2013/14. Universität Freiburg, Germany, 2013. Available at: <http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/> // Accessed 2016-12-08.
10. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. *Eighteenth National Conference on Artificial Intelligence; American Association for Artificial Intelligence: Menlo Park, CA, USA*, 2002; pp. 593–598.
11. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI); IJCAI*, 2003.
12. Siegwart, R.; Nourbakhsh, I.R.; Scaramuzza, D. *Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents series)*; The MIT Press, 2011.
13. Grisetti, G.; Kummerle, R.; Stachniss, C.; Burgard, W. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine* **2010**, *2*, 31–43.
14. Cheein, F.A.; Steiner, G.; Paina, G.P.; Carelli, R. Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection. *Computers and Electronics in Agriculture* **2011**, *78*, 195–207.
15. Hiremath, S.A.; van der Heijden, G.W.; van Evert, F.K.; Stein, A.; ter Braak, C.J. Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. *Computers and Electronics in Agriculture* **2014**, *100*, 41–50.
16. Lepej, P.; Rakun, J. Simultaneous localisation and mapping in a complex field environment. *Biosystems Engineering* **2016**, *150*, 160–169.

17. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality; IEEE Computer Society: Washington, DC, USA, 2007; ISMAR '07, pp. 1–10.
18. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A versatile and accurate monocular SLAM aystem. *IEEE Transactions on Robotics* **2015**, *31*, 1147–1163.
19. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. 2011 International Conference on Computer Vision, 2011, pp. 2564–2571.
20. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. European Conference on Computer Vision (ECCV), 2014, pp. 834–849.
21. Engel, J.; Stücker, J.; Cremers, D. Large-scale direct SLAM with stereo cameras. Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, 2015, pp. 1935–1942.
22. Caruso, D.; Engel, J.; Cremers, D. Large-scale direct SLAM for omnidirectional cameras. Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, 2015, pp. 141–148.
23. Gálvez-López, D.; Tardós, J.D. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics* **2012**, *28*, 1188–1197.
24. Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. g2o: A general framework for graph optimization. IEEE International Conference on Robotics and Automation, 2011, pp. 3607–3613.
25. Scaramuzza, D.; Martinelli, A.; Siegwart, R. A toolbox for easily calibrating omnidirectional cameras. Proc. of the IEEE International Conference on Intelligent Systems, IROS06, Beijing, China, 2006.
26. Takasu, T.; Yasuda, A. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. International Symposium on GPS/GNSS, International Convention Center Jeju; , 2009.
27. Takasu, T. RTKLIB: An Open Source Program Package for GNSS Positioning, 2013.
28. Bouguet, J.Y. Camera calibration toolbox for Matlab, 2008.
29. Mei, C.; Rives, P. Single View Point Omnidirectional Camera Calibration from Planar Grids. Proceedings 2007 IEEE International Conference on Robotics and Automation, 2007, pp. 3945–3950.

PUBLICATIONS IN THE DISSERTATION
SERIES

1. Berg Marklund, Björn (2013) Games in formal educational settings: obstacles for the development and use of learning games, Informatics. Licentiate Dissertation, ISBN 978-91-981474-0-72.
2. Aslam, Tehseen (2013) Analysis of manufacturing supply chains using system dynamics and multi-objective optimization, Informatics. Doctoral Dissertation, ISBN 978-91981474-1-43.
3. Laxhammar, Rikard (2014) Conformal Anomaly Detection-Detecting Abnormal Trajectories in Surveillance Applications, Informatics. Doctoral Dissertation, ISBN 978-91-981474-2-14.
4. Alklind Taylor, Anna-Sofia (2014) Facilitation matters: a framework for instructor-led serious gaming, Informatics. Doctoral Dissertation, ISBN 978-91-981474-4-55.
5. Holgersson, Jesper (2014) User participation in public e-service development: guidelines for including external users, Informatics. Doctoral Dissertation, ISBN 978-91-981474-5-26.
6. Kaidalova, Julia (2015) Towards a Definition of the role of Enterprise Modeling in the Context of Business and IT Alignment, Informatics. Licentiate Dissertation, ISBN 978-91-981474-6-97.
7. Rexhepi, Hanife (2015) Improving health care information systems – A key to evidence based medicine, Informatics. Licentiate Dissertation, ISBN 978-91-981474-7-68.
8. Berg Marklund, Björn (2015) Unpacking Digital Game-Based Learning: The complexities of developing and using educational games, Informatics. Doctoral Dissertation, ISBN 978-91-981474-8-3.
9. Fornlöf, Veronica (2016) Improved remaining useful life estimations for on-condition parts in aircraft engines, Informatics. Licentiate Dissertation, ISBN 978-91-981474-9-0.
10. Ohlander, Ulrika (2016) Towards Enhanced Tactical Support Systems, Informatics. Licentiate Dissertation, ISBN 978-91-982690-0-0.
11. Siegmund, Florian (2016) Dynamic Resampling for Preference-based Evolutionary Multi-objective Optimization of Stochastic Systems, Informatics. Doctoral Dissertation, ISBN 978-91-982690-1-7.
12. Kolbeinsson, Johan (2016) Managing Interruptions in Manufacturing: Towards a Theoretical Framework for Interruptions in Manufacturing Assembly, Informatics. Licentiate Dissertation, ISBN 978-91-982690-2-4.
13. Sigholm, Ari (2016) Secure Tactical Communications for Inter-Organizational Collaboration: The Role of Emerging Information and Communications Technology, Privacy Issues, and Cyber Threats on the Digital Battlefield, Informatics. Doctoral Dissertation, ISBN 978-91-982690-3-1.
14. Brolin, Anna (2016) An investigation of cognitive aspects affecting human performance in manual assembly, Informatics. Doctoral Dissertation, ISBN 978-91-982690-4-8.
15. Brodin, Martin (2016) Mobile device strategy: A management framework for securing company information assets on mobile devices, Informatics. Licentiate Dissertation, ISBN 978-91-982690-5-5.



STEFAN ERICSON

Stefan Ericson received his M.Sc. in Electrical Engineering from University of Skövde in 1999. Since 2002 he has been working as a lecturer at University of Skövde.

In this thesis Stefan investigates how cameras can be used for localization of an agricultural mobile robot. He focuses on relative measurement that can be used to determine where a weeding tool is operating relative a weed detection sensor. It incorporates downward-facing perspective cameras, forward-facing perspective cameras and omnidirectional cameras. Stefan shows how the camera's ego-motion can be estimated to obtain not only the position in 3D but also the orientation. He also shows how line structures in the field can be used to navigate a robot along the rows.